Universitetet i Oslo
Institutt for informatikk

Implementasjon av datavarehus – et eksempel

Hovedoppgave

Anette Langmo Becker

22. oktober 2004

Forord

Denne oppgaven er skrevet som en del av Cand. Scient.-graden ved Institutt for Informatikk ved Universitetet i Oslo. For å ha utbytte av å lese denne oppgaven bør leseren ha noe kjennskap til databaser og databasesystemer.

Jeg vil gjerne takke min veileder Ragnar Normann for konstruktiv veiledning. Det har vært inspirerende å ha møter med deg, og du har stilt opp til tross for at du har vært sykmeldt i lange perioder.

Takk til Knut Hegna som har gitt meg tilgang til kildefilene jeg har benyttet i denne oppgaven, og hjulpet meg å sette meg inn i disse.

Takk til Mamma og Pappa som alltid har stilt opp for meg og hatt troen på meg hele veien.

Takk til medstudenter for faglige innspill på veien.

Aller mest vil jeg takke min samboer, Erik, som har støttet meg og gitt meg oppmuntringer. Det har vært fint og alltid ha en å snakke med. Du har fulgt med og kommet med gode innspill underveis og du har vært en ypperlig korrekturleser. Tusen takk!

Blindern, oktober 2004
Anette Langmo Becker

Innhold

Figurer

Tabeller

1 Innledning og bakgrunn

I løpet av min studietid på Blindern, er databaser det fagområdet som har fascinert meg mest. Jeg bestemte meg tidlig for at min hovedoppgave skulle omhandle dette temaet. Et datavarehus er en spesiell type database, og datavarehusteknologien er et forholdsvis nytt område innenfor databaseverden. Datavarehus er et spennende tema som jeg ville lære mer om; jeg valgte derfor å skrive hovedoppgave om dette temaet.

Databaser er en samling av relaterte data. I tillegg til å være en samling av informasjon, er datavarehus et støttesystem. Datavarehus er integrerte data fra flere kilder lagret i en multidimensjonal modell. Hovedmålet med et datavarehus er at det skal være et verktøy til hjelp for beslutningsstøtte.

Fremgangen i analytiske verktøy og teknikker har resultert i utviklingen av datavarehus, [4]. Datavarehus har større lagringsplass, mer funksjonalitet og raskere responstider enn transaksjonsorienterte databaser.

Datavarehus er en teknologi som skal støtte beslutningstakere. Denne teknologien skal hjelpe bedrifter til å ta raskere og bedre avgjørelser. I følge [2] var det en stor vekst på dette området på midten av nittitallet. I denne perioden ble beslutningsstøtte et viktig tema. Utviklerne så på alle aspekter ved temaet, og datavarehus vokste fram som teknologi.

Gary Hallmark skrev i [6] i 1995 at datavarehus var et av de raskest voksende segmentene i dataindustrien. Markedet var estimert til å vokse fra $2 milliarder i 1995 til $6.9 milliarder i 1999. “The competitive advantage of consolidating and analyzing all corporate data will be so great that companies will not be able to afford to be without a data warehouse,” sier han. Ifølge [3] var datavarehusmarkedet estimert til å vokse fra $2 milliarder i 1995 til $8 milliarder i 1998. Datavarehus blir mer og mer brukt. Det er et nytt og spennende område.

1.1 Målsetning og problemstillinger

Gjennom arbeidet med denne oppgaven har målsetningen og problemstillingen utviklet og endret seg flere ganger:

Problemstilling 1 Hensikten med denne oppgaven var å definere og diskutere begrepene datavarehus og data mining. Videre skulle jeg vurdere hvilke databaseegenskaper som er relevante i denne forbindelse, og se på hvilke DBMSer som er egnet for slike anvendelser. Oppgaven skulle se på relasjonelle og objektorienterte databaseegenskaper og se hvilke egenskaper ved Relational DataBase Management System (RDBMS) og Object Oriented DataBase Management System (OODBMS) som er viktige og relevante for datavarehus og data mining. Ved implementasjon av samme datavarehus

på en relasjonell og en objektorientert database, skulle jeg studere ulike egenskaper ved disse.

Versjon 1 av problemstilling 2 Vi hadde ikke noe datavarehus tilgjengelig, derfor skulle jeg implementere to for å ha noe og teste på. Etter hvert som det viste seg at den objektorienterte delen av oppgaven uteble på grunn av tidspress, ble formålet med oppgaven endret. Målet ble da å implementere et datavarehus på en relasjonell database og se på størrelse og responstider på disse. Tar databasen eller datavarehuset størst plass, og hvem er det raskest å spørre mot?

Versjon 2 av problemstiling 2 Da den relasjonelle dokumentdatabasen var ferdig, skulle datavarehuset lages ut i fra denne. Det viste seg da at dokumentdatabasen hadde for mange nullverdier. Den var ikke noe godt grunnlag for et datavarehus. Nye data måtte til. Med nye artikkeldata kunne datavarehuset implementeres. Oppgavens mål var fortsatt å se på størrelse og responstider på databasen versus datavarehuset; hvem tar størst plass, og hvem er det raskest å spørre mot?

Denne tredelte utviklingen i oppgaven er detaljert forklart i kapittel 3, 4 og 5.

1.2 Metode

Strategi for å løse oppgavens problemstilling har bestått av:

- Bakgrunnsstudier i datavarehus og data mining.
- Finne datagrunnlag for databasene som skulle implementeres.
- Forstå innholdet og strukturen i datagrunnlaget.
- Definere strukturen på en relasjonell database ut i fra datagrunnlaget.
- Implementasjon av en relasjonell database på bakgrunn av det opprinnelige datagrunnlaget.
- Finne nytt datagrunnlag.
- Implementasjon av en relasjonell database ut i fra det nye datagrunnlaget.
- Implementasjon av et datavarehus på den siste relasjonelle databasen.

- Testing og spørring mot databasen og datavarehuset.
- Evaluering og vurdering av resultatene.

1.3 Disposisjon av oppgaven

Oppgaven består av i alt 9 kapitler og kan grovt deles inn i tre.

Del 1 Innledning og begrepsinnføring Del 1 er ment å gi leseren en innledning til oppgaven og en innføring i noen sentrale begreper.

- Kapittel 1 inneholder en innledning og en kort oversikt over oppgaven.
- Kapittel 2 er en innføring av begrepet datavarehus.

Del 2 Implementering Del 2 omhandler selve implementeringen, og gir et bilde av den praktiske utførelsen av oppgaven.

- Kapittel 3 beskriver arbeidet med problemstilling 1.
- Kapittel 4 beskriver arbeidet med versjon 1 av problemstilling 2.
- Kapittel 5 beskriver arbeidet med versjon 2 av problemstilling 2.

Del 3 Resultater Del 3 viser resultatene av oppgaven.

- Kapittel 6 gir en oversikt over spørringene som er brukt i den praktiske utførelsen av oppgaven.
- Kapittel 7 viser resultatene fra kjøringene av spørringene.
- Kapittel 8 gir en konklusjon og avslutning.
- Kapittel 9 gir en oversikt over vedleggene til oppgaven.

Vedlegg A Javaprogrammet som leser inn den første relasjonelle databasen, ut i fra det opprinnelige datagrunnlaget.

Vedlegg B NIAM-analysen av den første relasjonelle databasen.

Vedlegg C Oracleskjema for den første relasjonelle databasen.

Vedlegg D Javaprogrammet som leser inn den andre relasjonelle databasen, ut i fra det nye datagrunnlaget.

Vedlegg E NIAM-analysen av den andre relasjonelle databasen.

Vedlegg F Oracleskjema for den andre relasjonelle databasen.

1.4 Språk og ordbruk

Jeg har valgt å skrive oppgaven på norsk. I dataverden brukes mange engelske begreper og termer. Ofte finnes det ikke noen god norsk oversettelse av disse termene. De ordene jeg mener har en god norsk oversettelse har jeg valgt å oversette, mens for de som ikke har det har jeg valgt å bruke de engelske termene.

Jeg har for eksempel valgt å bruke det engelske uttrykket for disse termene: join, select, view, insert, update, delete, read-only, roll-up og drill-down.

Ord som jeg har valgt å oversette er for eksempel:

- decision support som er oversatt med beslutningsstøtte
- query som er oversatt med spørsmål eller spørring
- fact table som er oversatt med faktatabell
- dimension table som er oversatt med dimensjonstabell
- datacube som er oversatt med datakube
- star schema som er oversatt med stjernediagram
- snowflake schema som er oversatt med snøfnuggdiagram
- fact constellation som er oversatt med faktagruppering
- pivoting som er oversatt med rotasjon

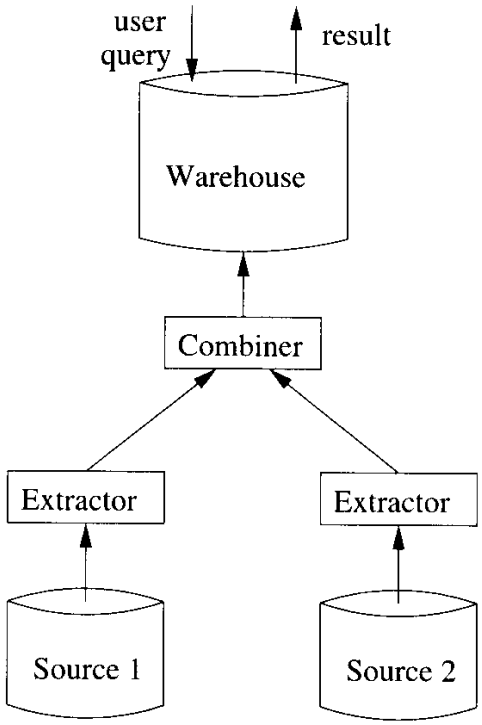
2 Datavarehus

Dette kapittelet gir en innføring i begrepet datavarehus.

2.1 Begrep

Jeg har valgt å bruke begrepsforståelsen fra [4] og [5].

Et datavarehus er en spesialutgave av en database. Kopier av data fra én eller flere kildedatabaser er lagret i én database, som kalles datavarehus, se figur 1. De ulike kildene kan ha forskjellige plattformer og lagringsmåter. Det vil si at dataene, som skal lagres i datavarehuset, ofte har ulik struktur i de ulike kildene. Dataene må da integreres og få et felles format før de kan lagres i datvarehuset.



Figur 1: Kopier av data fra en eller flere kildedatabaser bearbeides og lagres i et datavarehus. Fra [5].

Datavarehus har en enkel struktur, men er stort når det gjelder data. Selv om datavarehuset gjerne inneholder mer data enn de opprinnelige kildene, så går eksekveringen av spørsmål mot datavarehuset raskere. Det kommer av at data fra flere databaser allerede er joinet.

Et datavarehus har mye til felles med view i SQL. Man kan se på et datavarehus som en utvidelse av database views. Begge har read-only tilgang til databasen. I [4] nevnes følgende forskjeller:

1. Et datavarehus blir fysisk lagret, og det eksisterer dermed hele tiden. Et view er derimot bare en presentasjon.
2. Views i relasjonsdatabaser er relasjonelle, mens datavarehus ofte er flerdimensjonale.
3. Man kan bruke indekser for å optimalisere et datavarehus, mens et view ikke kan indekseres uavhengig av den underliggende databasen.
4. Et datavarehus inneholder integrerte data fra flere databaser, mens et view bare inneholder data fra én database.

Punkt 1 og 3 er jeg enig i. Punkt 4 er ikke opplagt. Datavarehus kan inneholde data fra bare én database, og et view kan inneholde data fra flere databaser. Men det som forekommer oftest vil nok være som punkt 4 sier, at datavarehus inneholder data fra flere databaser, og view fra én. Punkt 2 er jeg ikke enig i. Jeg mener at det ikke er noen motsetning mellom relasjonell og flerdimensjonal. Datavarehus kan være både relasjonelle og flerdimensjonale.

Når man bygger et datavarehus, må forventet bruk beskrives nøye. Men det er ikke mulig å forutse alle spørsmål på forhånd, så ifølge [9] skal designet takle ad-hoc spørringer. Ad-hoc spørsmål vil si at brukeren sender vilkårlige spørsmål, etter som det faller ham/henne inn, mot datavarehuset.

For å sikre gyldighet og kvalitet på dataene, må dataene «renses» («cleanes») før de settes inn i datavarehuset. Dataene må formateres konsistent.

Datavarehus er stabile (non-volatile). Det vil si at dataene ikke blir borte hvis for eksempel strømmen blir tatt. Det tilfredsstiller D-kravet (durability) i ACID-reglene.

2.2 Formål med datavarehus

Et datavarehus er et interaktivt støtteverktøy for beslutning. Brukeren trenger rask tilgang til store mengder data integrert fra ulike kilder. Med datavarehussystemer prøver man å minke responstiden ved spørsmål så mye som mulig.

I datavarehusapplikasjoner er det viktigere å ha rask tilgang til data enn å unngå anomalier. Man bryter normalformene med vilje for å få opp farten. Denormaliseringen i datavarehuset gir imidlertid ikke inkonsistens. Man oppdaterer bare fra normaliserte data.

2.3 Hvorfor datavarehus

Hvorfor lager man datavarehus? Kan man ikke bare spørre mot de opprinnelige kildedatabasene? Noen av grunnene til at datavarehus brukes er:

- Kildedatabasene har mange brukere, og har derfor ikke nok kapasitet.
- I datavarehuset et dataene integrert. Det vil si at de har fått felles format.
- Det kan være lagret data i datavarehuset som bare kan gjenfinnes som aggregerte data i kilden.

2.4 Brukere

Datavarehus er systemer som er dedikert for analyse. De er ment som verktøy for å ta raske og gode beslutninger. Ifølge [9] er resultatet av typiske spørsmål i datavarehus aggregerte data fra store mengder data, og ikke verdier av spesielle tupler. Typiske brukere er analytikere og beslutningstakere.

2.5 Datavarehus vs. transaksjonsorienterte systemer

Tradisjonelle databaser balanserer kravet til dataaksess med behovet for å sikre integritet av data. I datavarehusapplikasjoner er det viktigere å ha rask tilgang til data enn å sikre integritet.

Hovedmålet med transaksjonsorienterte databasesystemer (produksjonssystemer) er å reagere på ordre så fort og billig som mulig. Raske oppdateringer er viktig. Hovedmålet med et datavarehus er at det skal fungere som et ledelsesstøttesystem. Det er et verktøy for å fatte riktige beslutninger. Da er det ikke raske oppdateringer som er det viktigste, men raske svar.

Datavarehus er en leseomgivelse. Update og delete skjer ytterst sjelden, fordi dette vil føre til inkonsistens i forhold til kildene. I transaksjonsorienterte databasesystemer derimot, skjer det mye update.

En av fordelene med datavarehus er at man slipper å forsinke transaksjoner i databasen, fordi antall spørringer mot databasen blir redusert.

2.6 OLAP og OLTP

OLAP (On-Line Analytic Processing) og OLTP (On Line Transaction Processing) er applikasjoner som datavarehus kan støtte.

OLAP er et begrep som beskriver analysen av komplekse data fra datavarehuset, [4].

Organisasjoner lager datavarehus med en stor del av deres tilgjengelige data. Analytikere sender spørsmål mot datavarehuset for å finne mønstre og trender som er viktig for organisasjonen. Denne aktiviteten kalles for OLAP, [5].

OLAP involverer som regel komplekse spørsmål med én eller flere aggregeringer. Disse spørsmålene kalles ofte OLAP-spørsmål eller spørsmål for beslutningsstøtte (decision-support spørsmål). OLAP spørsmål berører store mengder data. De tar ofte for lang tid å kjøre i transaksjonssystemer. OLAP er lesing fra datavarehuset. Nesten alle datavarehus har denne funksjonen.

Tradisjonelle databaser støtter OLTP. OLTP inkluderer insert, update og delete. OLTP-spørsmål berører som regel bare en liten del av databasen, ofte bare ett eller noen få tupler. De omfatter altså små mengder data. Ikke alle datavarehus støtter OLTP. Ved OLTP oppdateres datavarehuset mens det er i bruk.

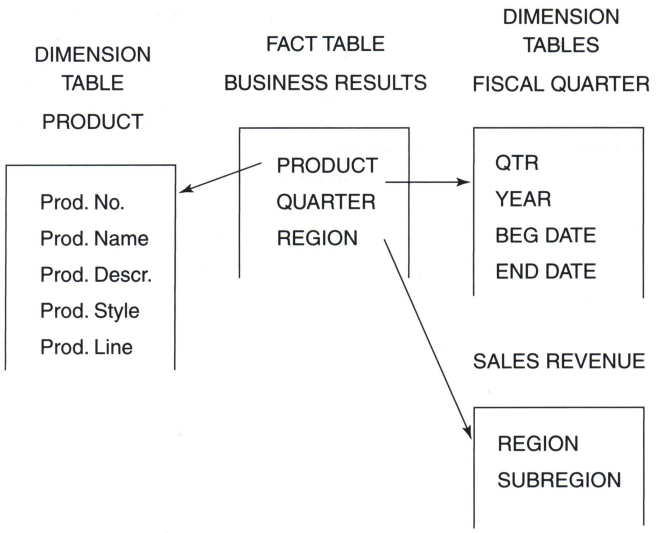
I en typisk OLAP-applikasjon er det en sentral relasjon eller samling av data som kalles faktatabell. En faktatabell representerer hendelser eller objekter av interesse. Attributtene i faktatabellen kan være pekere (fremmednøkler) til dimensjonstabeller (se forklaring under 2.7).

2.7 Datakuber

Den vanligste datamodellen for datavarehus er multidimensjonale matriser som kalles datakuber. En multidimensjonal lagringsmodell har to typer tabeller:

- Faktatabeller (se figur 2 og 3):
  - En forholdstabell som viser den sentrale informasjonen, for eksempel *Business results* i figurene.
  - Inneholder data fra kildedatabasene.
  - Data om hendelser som har skjedd, for eksempel salg, leveranser og produksjon.
  - Har en tid eller tidsperiode forbundet med seg.
  - Er store tabeller.
  - Data endres sjelden. Ingen update. Kan legge til nye attributter og tupler, men ikke endre de som allerede fins.
- Dimensjonstabeller (se figur 2 og 3):
  - Attributtene til én dimensjon av faktatabellen.
  - Pekere til disse fra faktatabellen. Eks: *Product*, *Fiscal quarter* og *Sales revenue* i figurene.

- Informasjon i dimensjonstabeller kan endres.
- Kan ha en kjede med dimensjonstabeller for å beskrive attributter i andre dimensjonstabeller.



Figur 2: Et stjernediagram består av en faktatabell med én tabell for hver dimensjon. Faktatabellen er i midten av stjerna, og dimensjonstabellene er rundt. Fra [4].

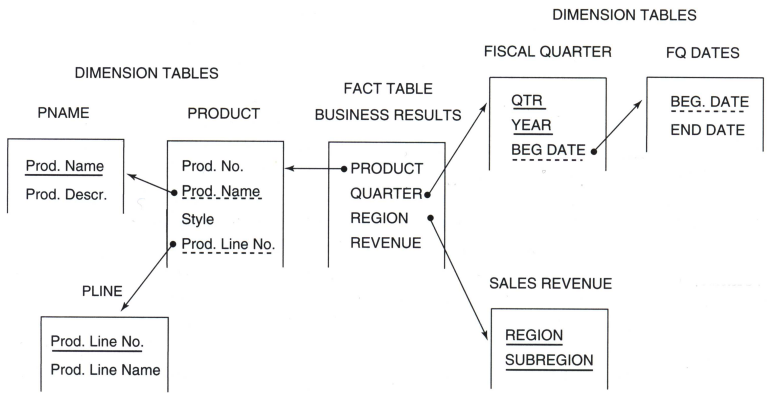
I [4] presenterer Elmasri og Navathe to typer multidimensjonale skjemaer:

- Stjernediagram (Star schema) består av en faktatabell med én tabell for hver dimensjon. Faktatabellen er i midten av stjerna, og dimensjonstabellene er rundt (se figur 2).
- Snøfnuggdiagram (Snowflake schema) er en variasjon av stjernediagram der dimensjonstabellene er organisert hierarkisk ved å normalisere dem (se figur 3).

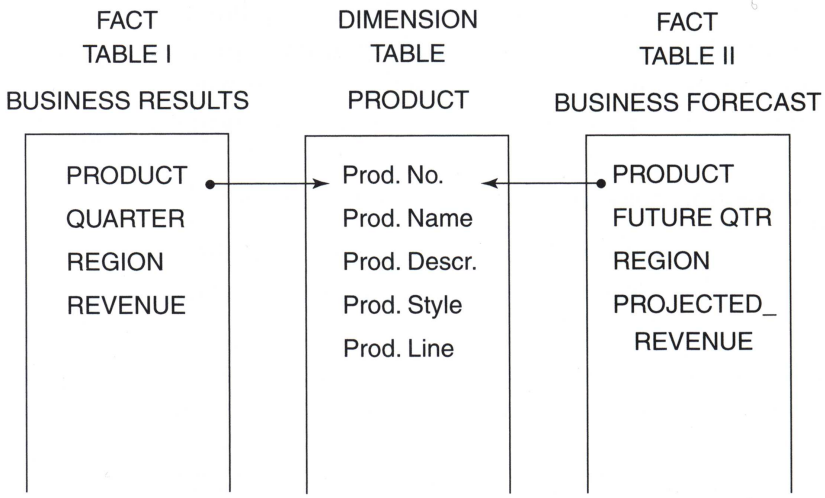
Flere faktatabeller kan dele samme dimensjonstabell. Dette kalles faktagruppering (fact constellation) (se figur 4).

Strukturen i et datavarehus er oppsummert i en datakube, se figur 5. Her er hver dimensjonstabell representert som en dimensjon i kuben. Man kan rotere kuben fra en dimensjonsorientering til en annen. Dette kalles rotasjon (pivoting).

Formelle datakuber aggregerer i alle retninger i alle dimensjoner, [5]. Disse aggregerte verdiene finnes i egne rader og kolonner i datakuben.



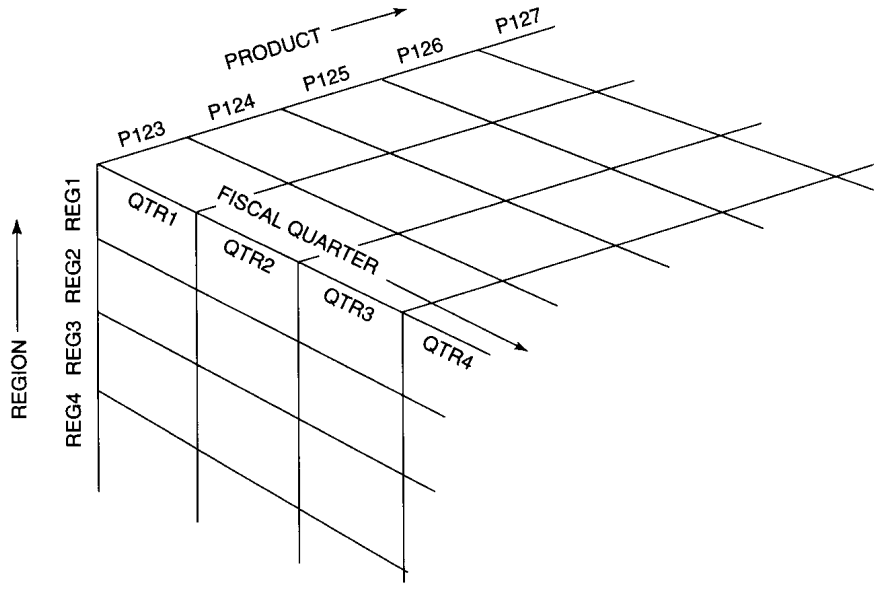
Figur 3: Snøfnuggdiagram er en variasjon av stjernediagram der dimensjonstabellene er organisert hierarkisk ved å normalisere dem. Fra [4].



Figur 4: Når flere faktatabeller deler samme dimensjonstabell, kalles det faktagruppering. Fra [4].

Ifølge [4] kan man foreta to operasjoner på en dimensjon:

- Roll-up beveger seg oppover i hierarkiet og grupperer i større gr-



Figur 5: Den multidimensjonale strukturen i et datavarehus er oppsummert i en datakube. Her er det en dimensjon for hver dimensjonstabell. Fra [4].

upper langs dimensjonen, for eksempel ved å summere ukentlige data på måned eller år.

- Drill-down fungerer motsatt. Det vil si at man grupperer i mindre grupper langs dimensjonen, for eksempel ved å dele opp en region i subregioner.

2.8 Problemer

Oppdatering av datavarehuset er vanskelig på grunn av størrelsen (mye data). Datavarehus er designet for leseaksess. Kvalitet og konsistente data er viktig.

Administrasjon av datavarehus krever mer kunnskaper og ferdigheter enn tradisjonell databaseadministrasjon. Et team med dyktige tekniske eksperter med ulik ekspertise trengs, heller enn én person. Administrasjonen er, som for databaser, bare delvis teknisk. Mye av administrasjonen er ansvaret for å jobbe med medlemmene av organisasjonen som har interesse av datavarehuset. Det trengs tekniske ferdigheter, nøye koordinasjon og effektivt lederskap.

Å få tilgjengelig kildedata inn i datamodellen til datavarehuset er en evig utfordring. Garcia-Molina, Ullman og Widom sier i [5] at det er minst

tre måter å oppdatere et datavarehus på:

- Periodisk rekonstruering. Det vil si at datavarehuset rekonstrueres i forhold til oppdaterte data i kildene. Dette skjer til bestemte tider, for eksempel hver natt. Dette er den mest brukte metoden. Problemet er at datavarehuset må kobles fra når oppdateringen skjer. Det er uheldig i de tilfellene hvor oppdateringen tar mer enn en natt. Dessuten kan dataene bli gamle i løpet av en hel dag.
- Periodisk oppdatering. Dette skjer også til bestemte tider, for eksempel hver natt. Datavarehuset oppdateres med de forandringene som er gjort i kildene siden forrige gang datavarehuset ble modifisert.
- Øyeblikkelig oppdatering. Datavarehuset oppdateres hver gang det skjer endringer i én eller flere av kildene.

Hvilken oppdateringsmetode man bør velge er diskutert i [3]. Det avhenger av kildene og mulighetene til databaseserverne.

2.9 Oppsummering - datavarehus vs. operasjonelle databaser

Målet med datavarehus er å gi brukeren et verktøy for interaktiv beslutningsstøtte. Brukeren trenger da rask tilgang til store mengder data integrert fra ulike kilder.

Administrasjon av datavarehus krever mer kunnskaper og ferdigheter enn tradisjonell databaseadministrasjon. Administrasjon av datavarehus er en stor jobb, proporsjonal med størrelsen og kompleksiteten på datavarehuset. Det er altså dyrt å lage datavarehus. Man trenger personer med riktig kompetanse. Behovet bør absolutt være til stede før man vurderer å lage et datavarehus; mye data og mange brukere kan indikere et behov. De fleste datavarehus har mer enn 1 000 brukere, og 30 prosent av datavarehusene er over 1 terabyte. Datavarehus inneholder mye data og er ofte mer enn dobbelt så store som kildene, [4].

Det er mer jobb å lage datavarehus i tillegg til de operasjonelle databasene, og det tar mer plass, men det frigjør kapasitet i databasene slik at databasene og datavarehuset går fortere og er mer effektive hver for seg til hver sine formål.

Beslutningsstøtte stiller andre krav til databaseteknologi enn tradisjonelle OLTP applikasjoner, [2, 3]. OLTP applikasjoner automatiserer typisk daglige operasjoner i en organisasjon. Oppgavene her består av korte, atomære og isolerte transaksjoner som krever oppdaterte (up-to-date) data og leser eller endrer noen få forekomster. Konsistens og backup av databasen er kritisk, det viktigste er systemets evne til å få unna transaksjonene. Beslutningsstøtte, derimot, krever historiske, summerte og kvalitetssikrede data fra mange kilder i hele bedriften. Siden

datavarehus inneholder kvalitetssikrede data, kanskje fra flere operasjonelle databaser, over en potensielt lang tidsperiode, pleier de å være mange ganger større enn operasjonelle databaser. Datavarehus i bedrifter kan være fra noen hundre gigabyte til flere terabyte i størrelse. Spørringene er ad-hoc og komplekse og kan aksessere millioner av forekomster.

I [7] ser vi at operasjonelle databaser inneholder tilstandsinformasjon, mens datavarehus typisk inneholder historisk informasjon. Et resultat av det er at datavarehus blir store og vokser over tid. Brukerne av beslutningsstøttesystemer er typisk interessert i å identifisere trender heller enn å se på individuelle forekomster isolert. Beslutningsstøttespørsmål er derfor mer komplekse enn OLTP spørsmål. Dessuten bruker de mer aggregering.

Applikasjoner for beslutningsstøtte involverer komplekse spørsmål mot store databaser. Siden responstidene helst skal være korte, er spørringsoptimalisering kritisk. Kravene til responstid er noen få sekunder eller toppen noen få minutter, [7]. Det er flere måter å få ned responstiden på. Bruk av indekser er vanlig, dessuten er det en kjent teknikk å beregne ofte spurte spørsmål på forhånd. Problemet er å definere hvilke spørsmål som er mest hensiktsmessige å forhåndsberegne, og begrense mengden av disse. Det blir for dyrt å beregne alle mulige spørsmål. [13] tar opp dette problemet med konfigurasjon av datavarehus. Behovet for lave spørsmålsprosesseringskostnader er i konflikt med behovet for lav forhåndsberegningskostnad.

Et spørsmål i forbindelse med utviklingen av datavarehus er hvilke dimensjoner som skal være med. I [10] ser vi hvordan dette var for Boeing Company. Hvert fly har potensielt en million attributter. Det er umulig å utvikle en multidimensjonal kube som har en million dimensjoner. Det sentrale spørsmålet blir da: Hva er fortjenesten med å introdusere en dimensjon til og hva blir kostnaden? Å legge til en dimensjonstabell i stjernediagrammet vil si å legge til en tabell i joinspørringen som lager faktatabellen. Kostnaden med å fremstille faktatabellen øker da eksponensielt.

3 Praktisk utførelse - problemstilling 1

Dette kapittelet gir en beskrivelse av arbeidet rundt den første problemstillingen.

3.1 Opprinnelig problemstilling

Oppgavens tittel er "Implementasjon av datavarehus – et eksempel." Ved implementasjon av samme datavarehus på en relasjonell og en objektorientert database, var det opprinnelige målet å finne hvilke egenskaper ved RDBMS og OODBMS som er viktige og relevante for datavarehus og data mining.

3.2 Kildefilen

Vi hadde ikke noe datavarehus å teste på, så første del av oppgaven var å implementere en relasjonell og en objektorientert database og et datavarehus på hver av dem. Databasene måtte selvfølgelig fylles med data. Som kildefil til databasene ble valgt filen ifibib.bsy. Filen inneholder informasjon om alle dokumenter (bøker, tidsskrifter, artikler osv.) ved biblioteket til Institutt for Informatikk ved Universitetet i Oslo. Filen er på 40 MB, og inneholder informasjon om 18 259 dokumenter. Strukturen i filen er vist i figur 6 på neste side.

Figuren viser de to første dokumentene i kildefilen. En ^ betyr at nå kommer det et nytt dokument. All informasjon om et dokument ligger altså samlet mellom to ^-er. *xxx er koder som skiller mellom ulike typer informasjon. Her er en oversikt over de kodene jeg har funnet relevante å ta med i databasene:

- *001 identifikasjonsnummer på dokumentet
- *008 ulike informasjonskoder
- *012 lokal informasjon
- *014 leverandørinformasjon
- *015 andre bibliografiske kontrollnummer
- *020 ISBN
- *022 ISSN
- *041 språkkoder
- *062 CR klassifikasjonsnummer
- *097 felles leverandørbeholdning

```
^
*000  pam         1
*001    0000000
*008  981111
*015  $a982423640
$bbibsys
*24510$a2d API
*260  $cc1999
*492  $vVol.3
*850  $n00ni00784$aUMN/INF
*014  $a akad
*999  $a umn/inf
*012  $a ifi
^
*000  pam         1
*001    0000001
*008  941021                  t          eng
*015  $a942695259
$bbibsys
*022  $a1078-2192
*24510$a3c on-line
$ba quarterly publication of the Association for Computing
 Machinery, Special Interest Group for Computing at
 Community Colleges
$w3c online
*260  $aNew York
$bAssociaton for Computing Machinery
*500  $aKvartalsvis
*500  $a1(1994) komplett med 1 hefte.
*500  $aOpphørt som trykt publikasjon med: 4(1997)nr 4,
 fortsettelsen finnes kun som nettdokument
*687  $aacm
*687  $atidsskrifter
*687  $ainformatikk
*710  $aAssociation for Computing Machinery
$bSpecial Interest Group for
 Computing at Community Colleges
*850  $n94ni00831$aUMN/INF
$b Tidsskr.
$c 3c ...
$g 1(1994)-4(1997)
*014  $a nic
*999  $a umn/inf
*012  $a ubni
^
```

Figur 6: Strukturen på den opprinnelige kildefilen, ifibib.bsy

- *250 utgave
- *260 utgivelse og distribusjon
- *300 fysisk beskrivelse
- *310 periodisitet
- *500 generell note
- *650 generelle emneord
- *653 frie nøkkelord
- *687 lokale kontrollerte emneord
- *850 eiende institusjon og beholdningsinfo
- *856 elektronisk lokalisering og tilgang

Det skilles mellom hovedordningsord, emneinnførsel, biinnførsel og serieinnførsel på et dokument. Dessuten kan ansvarlige for et dokument enten være en *person*, en *korporasjon* eller en *konferanse* (møte, utstilling og liknende). Dette gjenspeiles i kildefilen ved disse *xxx-kodene:

- *100 hovedinnførsel under person
- *110 hovedinnførsel under korporasjon
- *111 hovedinnførsel under konferanse
- *600 emneinnførsel under person
- *610 emneinnførsel under korporasjon
- *700 biinnførsel under person
- *710 biinnførsel under korporasjon
- *711 biinnførsel under konferanse
- *800 serieinnførsel under person

Tittel er også organisert under flere koder etter hvilken type tittel det er snakk om:

- *130 standardtittel
- *210 forkortet tittel
- *222 nøkkeltittel

- *240 standardtittel
- *245 tittel
- *246 paralelltittel
- *630 emneinnførsel under standardtittel
- *730 biinnførsel under standardtittel
- *740 biinnførsel tittel
- *830 serieinnførsel under standardtittel

Hver *xxx-kode har igjen et ulikt antall $-koder under seg. For eksempel er *014 delt opp i $-koder slik:

- $a leverandør-id
- $b leverandørnavn
- $c leverandøradresse
- $d postnummer og poststed
- $e land

Lagringsmåten i filen følger strenge internasjonale katalogiseringsregler. Disse reglene er beskrevet i [1] og [11]. MARC er akronym for MAchine-Readable Cataloging, og definerer et dataformat som stammer fra et Library of Congress-ledet initiativ som startet for over 30 år siden. Den nyeste versjonen av MARC kalles MARC 21. The Network Development and MARC Standards Office ved the Library of Congress og the Standards and the Support Office ved the National Library of Canada vedlikeholder MARC 21 formatet. NORMARC er en norsk standard for koding av bibliografiske opplysninger. BIBSYSMARC er en variant av MARC som brukes internt i Bibsys. Bibliotekene i Norge bruker ulike formater til å utveksle opplysninger. Kildefilen viser en slik måte å kode opplysninger på. Det finnes flere andre formater. For tiden pågår det en del arbeid med å representere slike data i xml, men noen standard er man ikke kommet fram til ennå.

3.3 Den første relasjonelle databasen

Den første relasjonelle databasen er implementert i Oracle. Java [8] ble brukt som programmeringsspråk for å legge informasjonen i kildefilen inn i databasen i Oracle, se vedlegg A. Databasen er stor og komplisert. Den inneholder 25 tabeller. NIAM-analysen kan sees i vedlegg B, og Oracleskjemaet kan sees i vedlegg C.

3.4 Kildefiler er ikke alltid enkle

Gjennom arbeidet med kildefilen har jeg smertelig fått erfare at kildefiler til datavarehus ikke alltid er enkle å jobbe med. Det var en stor jobb å sette seg inn i filens struktur og oppbygning. Kildefilen hadde mye overflødig informasjon i forhold til hva som var relevant å ta med i databasen. Etter å ha valgt ut hvilke deler av kildefilen som skulle være med, fant jeg en egnet struktur for databasen. Etter hvert som nye regler og begrensninger dukket opp var det stadig nødvendig å gå tilbake og endre NIAM-modellen og strukturen på databasen. Det var mye mer jobb enn ventet å sette seg inn i kildefilen. Den har en utrolig kompleks struktur. Jeg visste ikke at det lagres så mye informasjon om dokumentene på et bibliotek. Det var mye jobb og tok lang tid å gå gjennom alle *xxx-kodene i kildefilen og finne mulige underkoder ($-koder) for hver av dem. Dessuten skulle de ulike kodene lagres på riktig sted i Oracle-strukturen.

3.5 Endring i problemstilling

Strukturen på den relasjonelle databasen ble komplisert. Det var mye data som ikke skulle være med i datavarehuset. Dessuten hadde det tatt mye lengre tid enn planlagt å implementere den første databasen. I samråd med min veileder fant jeg ut at den objektorienterte delen av oppgaven ville ta for lang tid å gjennomføre. Det ville bli alt for mye arbeid. Dette ble oppdaget noe sent fordi min veileder var sykmeldt en lang periode.

4 Praktisk utførelse - versjon 1 av problemstilling 2

Dette kapittelet gir en beskrivelse av arbeidet rundt den første versjonen av den andre problemstillingen.

4.1 Ny problemstilling

Den objektorienterte delen av oppgaven falt altså bort fordi den ville tatt for mye tid. Oppgaven fikk derfor en ny vinkling. Den nye problemstillingen ble slik: Ut i fra den relasjonelle databasen og et datavarehus på den, hvor mye tid sparer man og hvor mye plass taper man på å ha et datavarehus? Altså: Hvor stor lagringsplass tar datavarehuset i forhold til databasen, og hvor lang tid tar det å spørre mot datavarehuset i forhold til mot databasen? Den objektorienterte delen, delen om data mining og delen om hvilke egenskaper ved RDBMS og OODBMS som er viktige og relevante for datavarehus og data mining utgikk. Oppgaven ble mer spesifikt rettet inn på ett eksempel ved implementasjon og om resultatene her ville bli som forventet. Tar virkelig datavarehuset større plass, og i såfall hvor mye? Går det virkelig raskere å spørre mot datavarehuset, og i tilfelle hvor mye raskere?

4.2 Utfordringer på veien

Prosessen med å legge dataene fra kildefilen over i en relasjonell database i Oracle gikk ikke helt knirkefritt. Jeg møtte en del utfordringer på veien. En ting som kan være verdt å nevne er “fnuttproblematikken” i Oracle. Hver gang jeg prøvde å legge inn en streng i Oracle fikk jeg nemlig feilmeldingen "Missing comma". Attributter av typen number gikk derimot helt greit. Det viste seg at alle strenger som sendes til Oracle skal ha fnutter (') rundt seg. Da gikk alt mye bedre helt til programmet prøvde å legge inn en streng med en fnutt midt inne i seg, for eksempel tittelen: “Grunnskolens ressursbruk - for n’te gang”. Da dukket feilmeldingen "Missing comma"opp igjen. Løsningen da ble å skrive et program som bytter ut alle enkeltfnutter(') i kildefilen med to enkeltfnutter etter hverandre(''). Oracle tolker da dette som en enkeltfnutt.

I forbindelse med testingen av den relasjonelle databasen, dukket det opp en ny utfordring. Oracle skriver nemlig ut en strek for hvert tegn under overskriften til hvert attributt i tabellene. Det vil si at hvis et attributt er varchar2(300) så skriver Oracle ut 300 streker under attributtoverskriften. Det gjør at utskriften blir uleselig. [12] var til god hjelp her. Oracle har kommandoen *column* for å angi hvor mange tegn hvert attributt skal ha når det skrives ut til skjerm.

4.3 Ny endring i oppgaven

Da datavarehuset skulle lages ut i fra den relasjonelle databasen, viste det seg at dokumentdatabasen ikke egner seg til å lage datavarehus fra. Databasen har for mange nullverdier. I databasen er det lagret informasjon om de ulike forekomstene på informatikkbiblioteket, for eksempel bøker, tidsskrifter og artikler. Det er stor forskjell på informasjonen som blir lagret om de ulike forekomstgruppene. Dette gjør at de sentrale delene av datakuben er for forskjellige. Dataene i databasen blir inhomogene fordi noen av tuplene er null på noen attributter, mens andre tupler er null på andre attributter. I datakuben blir dette et dårlig grunnlag for de forhåndsberegnede aggregeringene.

5 Praktisk utførelse - versjon 2 av problemstilling 2

Dette kapittelet gir en beskrivelse av arbeidet rundt den andre versjonen av den andre problemstillingen.

5.1 Nytt innhold i problemstillingen

Den relasjonelle databasen som var implementert egnet seg altså ikke som grunnlag for et datavarehus. På nytt ble oppgaven endret. Problemstillingen forble den samme, men nye data måtte til for å gjennomføre oppgaven. Det vil si at en ny database måtte implementeres som grunnlag for datavarehuset. Jeg fikk tilgang til noen filer med informasjon om artikler fra Nasjonalbiblioteket. Filene er lagret på samme format som ifibib.bsy, men inneholder litt annen informasjon. Blant annet har de informasjon om deweykoden for dokumentet. Denne nye informasjonen gjør at det blir en ekstra dimensjon i dataene. Filene inneholder data om samme type forekomster, nemlig artikler, og de har nesten ikke nullverdier. Dataene er derfor et bedre grunnlag for et datavarehus.

5.2 Ny relasjonell database

Som kildefiler for den nye relasjonelle databasen ble valgt de tre filene norart2000.mrc, norart2001.mrc og norart2002.mrc. Disse filene inneholder informasjon om til sammen 51 517 artikler som er utgitt i årene 2000, 2001 og 2002. Filene er mellom 5 og 6 MB hver. Filene har lik struktur. Strukturen kan sees i figur 7 på neste side. Figuren viser de to første dokumentene i norart2000.mrc. Tegnene ^, * og $ betyr det samme som før. *Person*, *Korporasjon* og *Tittel* er delt inn på samme måte som før, mens *Konferanse* ikke er med i disse filene. Her er en oversikt over de kodene jeg har funnet relevante å ta med i databasen:

- *001 identifikasjonsnummer på dokumentet
- *008 ulike informasjonskoder
- *041 språkkoder
- *082 Dewey desimalklassifikasjon
- *100 hovedinnførsel under person
- *110 hovedinnførsel under korporasjon
- *245 tittel
- *300 fysisk beskrivelse

```
^
*0010000270470
*008000131                              nor
*08230$a363.125
*100  $aMorland, Ellen
*245  $aFalsk trygghet
*300  $a[8]-9 : ill.
*520  $aOm bruk av mobiltelefon i bil med handsfreesett
*653  $amobiltelefoner
*653  $abilkj|ring
*773  $tForbruker-rapporten$gnr 1$i2000$x0046-449X
^
*0010000270472
*008000131                              nor
*08230$a338.43629283
*100  $aHattrem, Hanne
*245  $aPrisen p} lappen
*300  $a[10]-13 : ill.
*653  $af|rerkortpriser
*653  $akj|reskoler
*653  $aprisforskjeller
*773  $tForbruker-rapporten$gnr 1$i2000$x0046-449X
^
```

Figur 7: Strukturen på kildefilene norart2000.mrc, norart2001.mrc og norart2002.mrc. Figuren viser informasjon om to artikler.

- *520 sammendrag
- *600 emneinnførsel under person
- *610 emneinnførsel under korporasjon
- *653 frie nøkkelord
- *700 biinnførsel under person
- *773 vertsdokument

Strukturen på de nye kildefilene har mye til felles med strukturen på ifibib.bsy, som var kildefil til den første relasjonelle databasen. Mye av javaprogrammet som leste inn denne kunne derfor brukes videre. Dessuten kjente jeg til strukturen i filene, så det ble en enklere jobb å lese inn de nye filene enn det var å lese inn ifibib.bsy.

Etter en omskriving av javaprogrammet kunne altså de nye filene leses inn. Se vedlegg D for det nye javaprogrammet. For NIAM-analyse og Oracleskjema, se vedlegg E og F.

Den andre relasjonelle databasen ble mindre komplisert og mindre i størrelse enn den første relasjonelle databasen. Men den inneholder informasjon om flere enheter; mens den første databasen inneholder informasjon om 18 259 dokumenter, inneholder den andre informasjon om 51 517 artikler.

Den nye relasjonelle databasen har 9 tabeller og er på 72,5 MB. Det er til sammen 452 742 forekomster i tabellene. Fordelingen av forekomster per tabell kan sees i tabell 1. Databasen er optimalisert med indekser som ble automatisk generert av NIAM-suiten. Indeksene er hovedgrunnen til at databasen tar stor lagringsplass.

Tabell	**Antall forekomster**
Ans_Korp	5446
Ans_Pers	82845
Bokspraak	51661
Dokument	51517
Dok_Dew	46218
Korporasjon	5446
Nokkelord	102374
Person	82845
Sammendrag	24390

Tabell 1: Fordeling av forekomster i tabellene i den andre relasjonelle databasen.

5.3 Datavarehuset

Datavarehuset ble laget ved å velge ut data fra den andre relasjonelle databasen. Følgende tabeller ble tatt med: Ans_Korp, Ans_Pers, Bokspraak, Dokument, Dok_Dew og Nokkelord. Tabellene ble joinet i fem ulike spørringer der ønskede data fra én tabell ble lagt til for hver spørring, se figur 8 på neste side. Disse attributtene er med i datavarehuset:

- Id_V
- Reg_dato_V
- Dewey_V
- Spraak_V
- Pers_id_V
- Pers_kode_V
- Korp_id_V
- Korp_kode
- Nokkelord_V

Begge de relasjonelle databasene er optimalisert med indekser. Disse indeksene ble automatisk generert av NIAM-suiten. For å gjøre datavarehuset mer effektivt å spørre mot, ble det implementert en indeks på hvert attributt. Ulempen med indekser er at de tar mye plass. Uten indekser var datavarehuset på 11,5 MB. Med indekser er det på 42,6 MB. Datavarehuset inneholder 1 tabell og har 182 767 forekomster.

```
CREATE TABLE varehus_2 AS (
SELECT dok.Id_for AS Id_V, dok.Tid_reg_dato AS
Reg_dato_V, d.Deweykode_dew_kode AS Dewey_V
FROM abecker2.Dokument dok full outer join
abecker2.Dok_Dew d ON dok.Id_for = d.Id_dok);

CREATE TABLE varehus_3 AS (
SELECT Id_V, Reg_dato_V, Dewey_V, b.Spraak_brukt_i
AS Spraak_V
FROM varehus_2 full outer join abecker2.Bokspraak b
ON Id_V = b.Id_skrevet_paa);

CREATE TABLE varehus_4 AS (
SELECT Id_V, Reg_dato_V, Dewey_V, Spraak_V, p.P_Id_er
AS Pers_id_V, p.koden_for AS Pers_kode_V
FROM varehus_3 full outer join abecker2.Ans_Pers p
ON Id_V = p.Id_bok);

CREATE TABLE varehus_5 AS (
SELECT Id_V, Reg_dato_V, Dewey_V, Spraak_V, Pers_id_V,
Pers_kode_V, k.Korp_id_ansvarlig AS Korp_id_V,
k.Koden_for AS Korp_kode
FROM varehus_4 full outer join abecker2.Ans_Korp k
ON Id_V = k.Id_dok);

CREATE TABLE varehus_6 AS (
SELECT Id_V, Reg_dato_V, Dewey_V, Spraak_V, Pers_id_V,
Pers_kode_V, Korp_id_V, Korp_kode, n.Ordet_ord AS
Nokkelord_V
FROM varehus_5 full outer join abecker2.Nokkelord n
ON Id_V = n.Id_bok);
```

Figur 8: Kjøringene som lagde datavarehuset.

6 Spørringer

Jeg har laget ulike spørringer for å teste ut databasen og datavarehuset. Spørringene er varierte og plukker ut data fra alt fra en til seks tabeller i databasen. Dette mener jeg gir et godt grunnlag for å vurdere om databasen eller datavarehuset er mest lønnsomt å spørre mot.

6.1 Oversikt over spørringene

For å teste databasen mot datavarehuset ble disse spørringene kjørt mot databasen og datavarehuset:

6.1.1 Spørring 1

Hvor mange artikler er registrert på hvert nøkkelord?
Mot datavarehuset:

```
SELECT Nokkelord_V, Count(Nokkelord_V)
FROM varehus_6
GROUP BY Nokkelord_V;

Mot databasen:

SELECT Ordet_ord, Count(Ordet_ord)
FROM Nokkelord
GROUP BY Ordet_ord;
```

6.1.2 Spørring 2

Hvor mange prosent av artiklene er skrevet på hvert språk?
Mot datavarehuset:

```
SELECT Spraak_V,
(Count(Spraak_V)/(SELECT Count(Spraak_V)
FROM varehus_6))*100
FROM varehus_6
GROUP BY Spraak_V;

Mot databasen:

SELECT Spraak_brukt_i,
(Count(Spraak_brukt_i)/(SELECT Count(Spraak_brukt_i)
FROM Bokspraak))*100
FROM Bokspraak
GROUP BY Spraak_brukt_i;
```

6.1.3 Spørring 3

Hvor mange artikler er det innenfor et gitt emne?
Mot datavarehuset:

```
SELECT SUM(Count(Dewey_V))
FROM varehus_6
WHERE Dewey_V LIKE '1%'
GROUP BY Dewey_V;
```

Mot databasen:

```
SELECT SUM(Count(Deweykode_dew_kode))
FROM Dok_Dew
WHERE Deweykode_dew_kode LIKE '1%'
GROUP BY Deweykode_dew_kode;
```

6.1.4 Spørring 4

Hvor mange artikler er det innenfor hvert språk, gitt emne?
Mot datavarehuset:

```
SELECT Spraak_V, Count(Spraak_V)
FROM varehus_6
WHERE Dewey_V LIKE '4%'
GROUP BY Spraak_V;
```

Mot databasen:

```
SELECT b.Spraak_brukt_i, Count(b.Spraak_brukt_i)
FROM Bokspraak b, Dok_Dew d
WHERE b.Id_skrevet_paa = d.Id_dok
AND d.Deweykode_dew_kode LIKE '4%'
GROUP BY b.Spraak_brukt_i
```

6.1.5 Spørring 5

Hvor mange personer har skrevet på hvert språk?
Mot datavarehuset:

```
SELECT Spraak_V, Count(Pers_id_V)
FROM varehus_6
WHERE Pers_kode_V = '100'
GROUP BY Spraak_V;
```

Mot databasen:

```
SELECT b.Spraak_brukt_i, Count(a.P_Id_er)
FROM Bokspraak b, Ans_Pers a
WHERE a.Koden_for = '100'
AND b.Id_skrevet_paa = a.Id_bok
GROUP BY Spraak_brukt_i;
```

6.1.6 Spørring 6

Hvor mange personer har skrevet om gitt emne i 2002?
Mot datavarehuset:

```
SELECT Count(Pers_id_V)
FROM varehus_6
WHERE Dewey_V LIKE '1%'
AND Pers_kode_V = '100'
AND Reg_dato_V LIKE '02%';
```

Mot databasen:

```
SELECT Count(P_Id_er)
FROM Ans_Pers a, Dok_Dew d, Dokument dok
WHERE dok.Tid_reg_dato like '02%'
AND d.Deweykode_dew_kode LIKE '1%'
AND a.Koden_for = '100'
AND a.Id_bok = d.Id_dok
AND a.Id_bok = dok.Id_for
AND d.Id_dok = dok.Id_for;
```

6.1.7 Spørring 7

Hvor mange artikler er det på hvert språk i et gitt emne fra 2000?
Mot datavarehuset:

```
SELECT Spraak_V, Count(Id_V)
FROM varehus_6
WHERE Dewey_V LIKE '3%'
AND Reg_dato_V LIKE '02%'
GROUP BY Spraak_V;
```

Mot databasen:

```
SELECT b.Spraak_brukt_i, Count(b.Id_skrevet_paa)
FROM Bokspraak b, Dok_Dew d, Dokument dok
WHERE dok.Tid_reg_dato like '02%'
AND d.Deweykode_dew_kode LIKE '3%'
AND b.Id_skrevet_paa = d.Id_dok
AND b.Id_skrevet_paa = dok.Id_for
AND d.Id_dok = dok.Id_for
GROUP BY b.Spraak_brukt_i;
```

6.1.8 Spørring 8

Hvilke personer har skrevet artikler som ble gitt ut i 2001 gitt emne og språk?
Mot datavarehuset:

```
SELECT DISTINCT Pers_id_V
FROM varehus_6
WHERE Pers_kode_V = '100'
AND Dewey_V LIKE '3%'
AND Reg_dato_V LIKE '01%'
AND Spraak_V = 'dan'
ORDER BY Pers_id_V;
```

Mot databasen:

```
SELECT a.P_Id_er
FROM Ans_Pers a, Bokspraak b, Dok_Dew d, Dokument dok
WHERE a.Koden_for = '100'
AND dok.Tid_reg_dato like '01%'
AND d.Deweykode_dew_kode LIKE '3%'
AND b.Spraak_brukt_i = 'dan'
AND b.Id_skrevet_paa = d.Id_dok
AND b.Id_skrevet_paa = dok.Id_for
AND b.Id_skrevet_paa = a.Id_bok
AND d.Id_dok = dok.Id_for
AND d.Id_dok = a.Id_bok
AND dok.Id_for = a.Id_bok
ORDER BY a.P_Id_er;
```

6.1.9 Spørring 9

Hvilke nøkkelord har en gitt person skrevet om gitt år, emne og språk?
Mot datavarehuset:

```
SELECT Nokkelord_V
FROM varehus_6
WHERE Pers_id_V = '18010'
AND Dewey_V LIKE '3%'
AND Reg_dato_V LIKE '01%'
AND Spraak_V = 'dan';
```

Mot databasen:

```
SELECT n.Ordet_ord
FROM Nokkelord n, Ans_Pers a, Dok_Dew d,
Dokument dok, Bokspraak b
WHERE a.P_Id_er = '18010'
AND d.Deweykode_dew_kode LIKE '3%'
AND dok.Tid_reg_dato LIKE '01%'
AND b.Spraak_brukt_i = 'dan'
AND n.Id_bok = a.Id_bok
AND n.Id_bok = d.Id_dok
AND n.Id_bok = dok.Id_for
AND n.Id_bok = b.Id_skrevet_paa
AND a.Id_bok = d.Id_dok
AND a.Id_bok = dok.Id_for
AND a.Id_bok = b.Id_skrevet_paa
AND d.Id_dok = dok.Id_for
AND d.Id_dok = b.Id_skrevet_paa
AND dok.Id_for = b.Id_skrevet_paa;
```

6.1.10 Spørring 10

Hvilke artikler har gitte årstall, emne, språk, korporasjonid, personid, og nøkkelord?
Mot datavarehuset:

```
SELECT Id_V
FROM varehus_6
WHERE Reg_dato_V LIKE '00%'
AND Dewey_V LIKE '3%'
AND Spraak_V = 'nor'
AND Korp_id_V = '323'
AND Pers_id_V = '5094'
AND Nokkelord_V = '(IKT)';
```

Mot databasen:

```
SELECT dok.Id_for
FROM Dokument dok, Dok_Dew d, Bokspraak b,
Ans_Pers a, Ans_korp k, Nokkelord n
WHERE dok.Tid_reg_dato LIKE '00%'
AND d.Deweykode_dew_kode LIKE '3%'
AND b.Spraak_brukt_i = 'nor'
AND k.Korp_id_ansvarlig = '323'
AND a.P_Id_er = '5094'
AND n.Ordet_ord = '(IKT)'
AND dok.Id_for = d.Id_dok
AND dok.Id_for = b.Id_skrevet_paa
AND dok.Id_for = a.Id_bok
AND dok.Id_for = k.Id_dok
AND dok.Id_for = n.Id_bok
AND d.Id_dok = b.Id_skrevet_paa
AND d.Id_dok = a.Id_bok
AND d.Id_dok = k.Id_dok
AND d.Id_dok = n.Id_bok
AND b.Id_skrevet_paa = a.Id_bok
AND b.Id_skrevet_paa = k.Id_dok
AND b.Id_skrevet_paa = n.Id_bok
AND a.Id_bok = k.Id_dok
AND a.Id_bok = n.Id_bok
AND k.Id_dok = n.Id_bok;
```

7 Resultater

Spørringene fra kapittel 6 ble kjørt mot den andre relasjonelle databasen og mot datavarehuset med og uten indekser. Testplattformen er Red Hat Linux release 9 (Shrike) med 1 GB RAM, prosessor Intel(R) Pentium(R) 4 CPU 2.60GHz og cache størrelse 512 KB. Oracleversjon som er brukt er Oracle9i Enterprise Edition Release 9.2.0.4.0 med SQL Plus Release 8.1.7.0.0.

Tabell 2 på neste side viser resultatene fra kjøringene. Alle spørringene ble kjørt flere ganger, og tidene varierte med noen millisekunder. Mot databasen tok den første kjøringen av hver spørring mye lengre tid enn de andre kjøringene. Dette skyldes at skjemainformasjonen for tabellene må leses inn for hver spørring. Etter første kjøring finnes informasjonen i minnet. Siden resultatet fra den første kjøringen avviker så mye fra de andre kjøringene i databasen, har jeg valgt å utelate den første kjøringen fra gjennomsnittet. Den første kolonnen i tabell 2 på neste side er altså ikke med i utregningen av gjennomsnittet. I tabellen står **q1**, **q2** osv. for de ti ulike spørringene. Kjøringer mot databasen er betegnet med **db**, **dv** er kjøringer mot datavarehuset uten indekser og **dvi** er kjøringer mot datavarehuset med indeks. Alle tidene er i millisekunder.

7.1 Oversikt over resultatene fra spørringene

Her er en oversikt over de gjennomsnittlige tidsresultatene fra kjøringene:

1. Hvor mange artikler er registrert på hvert nøkkelord? Spørring 1 tok gjennomsnittlig:
   - 220,2 millisekunder mot databasen.
   - 373,6 millisekunder mot datavarehuset uten indekser.
   - 386 millisekunder mot datavarehuset med indekser.
2. Hvor mange prosent av artiklene er skrevet på hvert språk? Spørring 2 tok gjennomsnittlig:
   - 47,4 millisekunder mot databasen.
   - 207,6 millisekunder mot datavarehuset uten indekser.
   - 274,4 millisekunder mot datavarehuset med indekser.

   Spørring 1 og 2 gikk raskere mot databasen enn mot datavarehuset fordi spørringene bare plukket ut data fra en tabell. Tabellen spørringene ble kjørt mot er mindre i databasen enn i datavarehuset. Mot datavarehuset gikk spørringene raskest uten indekser. Det

	1	2	3	4	5	6	Gj.snitt
q1-db	552	216	234	214	212	225	220,2
q1-dv	458	389	371	372	367	369	373,6
q1-dvi	389	386	384	389	385	386	386
q2-db	266	48	49	45	47	48	47,4
q2-dv	199	202	206	206	217	207	207,6
q2-dvi	382	254	282	296	267	273	274,4
q3-db	243	16	16	15	15	16	15,6
q3-dv	66	64	65	70	64	67	66
q3-dvi	9	7	6	6	7	7	6,6
q4-db	327	24	25	24	24	25	24,4
q4-dv	62	61	60	61	59	60	60,2
q4-dvi	26	15	15	15	15	15	15
q5-db	743	456	452	453	451	452	452,8
q5-dv	117	108	110	107	111	110	109,2
q5-dvi	162	139	138	137	139	137	138
q6-db	743	184	186	184	184	184	184,4
q6-dv	74	72	71	74	74	73	72,8
q6-dvi	95	89	93	102	89	92	93
q7-db	229	226	226	225	225	227	225,8
q7-dv	92	82	81	83	82	84	82,4
q7-dvi	139	133	132	131	134	133	132,6
q8-db	444	16	16	16	16	16	16
q8-dv	78	58	56	56	59	61	58
q8-dvi	27	24	24	24	24	25	24,2
q9-db	24	3	3	3	3	3	3
q9-dv	63	62	61	61	62	61	61,4
q9-dvi	5	2	2	2	3	2	2,2
q10-db	52	3	4	4	3	3	3,4
q10-dv	69	60	57	59	57	60	58,6
q10-dvi	5	3	3	3	3	3	3

Tabell 2: Tidsresultater fra kjøringene. Tidene er i millisekunder. **q1**, **q2** osv. er de ti ulike spørringene, **db** er kjøringer mot databasen, **dv** er kjøringer mot datavarehuset uten indekser og **dvi** er kjøringer mot datavarehuset med indekser.

er overraskende. En mulig forklaring er at indeks på tabellen her gjør at datavarehuset blir større og mer komplisert å spørre mot. Indeksen må leses, og det tar tid.

3. Hvor mange artikler er det innenfor et gitt emne? Spørring 3 tok gjennomsnittlig:

   - 15,6 millisekunder mot databasen.
   - 66 millisekunder mot datavarehuset uten indekser.
   - 6,6 millisekunder mot datavarehuset med indekser.

   Spørring 3 gikk raskere mot databasen enn mot datavarehuset uten indekser fordi spørringene bare plukket ut data fra en tabell. Tabellen spørringen ble kjørt mot er mindre i databasen enn i datavarehuset. Indeks var avgjørende her. Det gikk mye raskere å slå opp deweykodene i datavarehuset med indekser.

4. Hvor mange artikler er det innenfor hvert språk, gitt emne? Spørring 4 tok gjennomsnittlig:

   - 24,4 millisekunder mot databasen.
   - 60,2 millisekunder mot datavarehuset uten indekser.
   - 15 millisekunder mot datavarehuset med indekser.

   Spørring 4 er en join mellom to tabeller i databasen. At spørringen stort sett gikk raskere mot databasen enn mot datavarehuset uten indeks skyldes at den plukker ut alle deweykoder som begynner med 4 i tabellen Dok_Dew i databasen før den joiner. Joinen blir da liten og tar kort tid. I datavarehuset uten indeks må den derimot gå igjennom alle forekomstene. I datavarehuset med indeks gikk det mye raskere å slå opp deweykodene.

5. Hvor mange personer har skrevet på hvert språk? Spørring 5 tok gjennomsnittlig:

   - 452,8 millisekunder mot databasen.
   - 109,2 millisekunder mot datavarehuset uten indekser.
   - 138 millisekunder mot datavarehuset med indekser.

   Spørring 5 er en join mellom to tabeller i databasen. At spørringen gikk raskere mot datavarehuset enn mot databasen skyldes at joinen i databasen er dyrere enn å gå gjennom alle elementene i datavarehuset. Indekser i datavarehuset gjorde ikke spørringen mer effektiv her. En mulig forklaring her er at det tar for lang tid å lese indeksene.

6. Hvor mange personer har skrevet om gitt emne i 2002? Spørring 6 tok gjennomsnittlig:

   - 184,4 millisekunder mot databasen.
   - 72,8 millisekunder mot datavarehuset uten indekser.
   - 93 millisekunder mot datavarehuset med indekser.

7. Hvor mange artikler er det på hvert språk i et gitt emne fra 2000? Spørring 7 tok gjennomsnittlig:

   - 225,8 millisekunder mot databasen.
   - 82,4 millisekunder mot datavarehuset uten indekser.
   - 132,6 millisekunder mot datavarehuset med indekser.

   Spørring 6 og 7 er en join mellom tre tabeller i databasen. Det tar derfor lengre tid å spørre mot databasen enn mot datavarehuset, der joinen allerede er foretatt. Det er overraskende at ikke indeksene i datavarehuset gjorde at spørringene her gikk raskere. Gitt emne og år burde favorisere indekser.

8. Hvilke personer har skrevet artikler som ble gitt ut i 2001 gitt emne og språk? Spørring 8 tok gjennomsnittlig:

   - 16 millisekunder mot databasen.
   - 58 millisekunder mot datavarehuset uten indekser.
   - 24,2 millisekunder mot datavarehuset med indekser.

   Mot databasen joiner denne spørringen fire tabeller, mens i datavarehuset er disse tabellene ferdig joinet. Forventet skulle spørringen derfor tatt lengre tid mot databasen enn mot datavarehuset. Derimot tok den stort sett lengre tid mot datavarehuset. Dette skyldes at nøkkelordet distinct må brukes mot datavarehuset for å få med hver person bare en gang i svaret. Distinct er altså dyrere enn join. Med indekser i datavarehuset går spørringen raskere, men databasen er fortsatt raskest fordi den ikke trenger nøkkelordet distinct.

9. Hvilke nøkkelord har en gitt person skrevet om gitt år, emne og språk? Spørring 9 tok gjennomsnittlig:

   - 3 millisekunder mot databasen.
   - 61,4 millisekunder mot datavarehuset uten indekser.
   - 3 millisekunder mot datavarehuset med indekser.

10. Hvilke artikler har gitte årstall, emne, språk, korporasjonid, personid, og nøkkelord? Spørring 10 tok gjennomsnittlig:

- 3,4 millisekunder mot databasen.
- 58,6 millisekunder mot datavarehuset uten indekser.
- 3 millisekunder mot datavarehuset med indekser.

Mot databasen joiner spørring 9 og 10 henholdsvis fem og seks tabeller, mens i datavarehuset er disse tabellene ferdig joinet. Forventet skulle spørringene derfor tatt lengre tid mot databasen enn mot datavarehuset. Dette stemmer når datavarehuset er optimalisert med indekser. Uten indekser tar det derimot mye tid å gå gjennom mange av elementene flere ganger.

8 Konklusjon

Dette kapittelet gir en oppsummering og avslutning på oppgaven.

8.1 Oppsummering

Temaet for oppgaven har vært datavarehus. Ved hjelp av et eksempel var målet å se hvor mye plass et datavarehus tar i forhold til den underliggende databasen. Dessuten ville jeg måle hvor lang responstid datavarehuset har i forhold til databasen.

8.2 Resultater

Lagringsplass og størrelse Størrelsene på databasen og datavarehuset er angitt i tabell 3.

Databasen inneholder 9 tabeller, er på 72,5 MB og har til sammen 452 742 forekomster.

Datavarehuset har 1 tabell og 182 767 forekomster. Uten indekser tar datavarehuset 11,5 MB, og med indekser tar det 42,6 MB.

Det vil si at databasen har 269 975 flere forekomster enn datavarehuset. Dessuten tar databasen 61 MB mer plass enn datavarehuset uten indekser og 29,9 MB mer plass enn datavarehuset med indekser. Indeksene i datavarehuset tar altså 31,1 MB lagringsplass. Indekser er også hovedgrunnen til at databasen bruker mye lagringsplass.

	ant tabeller	**lagringsplass**	**ant forekomster**
databasen	9	72,5 MB	452 742
datavarehuset	1	11,5/42,6 MB	182 767

Tabell 3: Lagringsplass og størrelse på datavarehuset og databasen.

Responstider De gjennomsnittlige tidsresultatene fra kjøringene er vist i tabell 4 på neste side. Alle tidene er i millisekunder.

Resultatene viser at spørringer som plukker ut data fra èn tabell i databasen, går raskere mot databasen enn mot datavarehuset. Dette skyldes at den ene tabellen det spørres mot, er mindre i databasen enn i datavarehuset. Hvis indeksene som er definert, er avgjørende for spørringen, vil spørringen gå raskere mot datavarehuset. Spørringer som plukker ut data fra to eller tre tabeller i databasen, går raskere mot datavarehuset enn mot databasen. Dette skyldes at spørringen må joine tabellene i databasen, mens i datavarehuset er tabellene allerede joinet. Riktig bruk

	Databasen	**Datavarehuset**	**Datavarehuset med indekser**
q1	220,2	373,6	386
q2	47,4	207,6	274,4
q3	15,6	66	6,6
q4	24,4	60,2	15
q5	452,8	109,2	138
q6	184,4	72,8	93
q7	225,8	82,4	132,6
q8	16	58	24,2
q9	3	61,4	2,2
q10	3,4	58,6	3

Tabell 4: Gjennomsnittlige tidsresultater fra kjøringene.

av indekser vil gjøre datavarehuset enda mer effektivt her. De tre siste spørringene plukker ut data fra henholdsvis fire, fem og seks tabeller i databasen. Disse spørringene gikk raskere mot databasen enn mot datavarehuset så lenge ikke nøkkelord som distinct må brukes i spørringen mot databasen. Dette skyldes at spørringene må gå gjennom mange av elementene i datavarehuset flere ganger. Det går raskere å plukke ut ønskede verdier i tabellene i databasen for deretter å foreta en liten join mellom disse. Når datavarehuset er optimalisert med indekser, er det derimot mer effektivt enn databasen her.

8.3 Konklusjon

Noen ulemper med å implementere et datavarehus i tillegg til de underliggende operasjonelle databasene:

- Det tar lang tid og krever mye ressurser.
- Det trengs mer lagringsplass.
- Jobben med å oppdatere datavarehuset er komplisert.

Noen fordeler med å implementere et datavarehus i tillegg:

- Det er mindre komplisert og tar kortere tid å skrive spørringene mot datavarehuset enn mot databasen.
- Dataene i datavarehuset er integrert, det vil si at de har et felles format.
- Det frigjøres kapasitet i kildedatabasene.

- Brukeren får raskt tilgang til store mengder data integrert fra ulike kilder.

En oppsummering av teorien fins i 2.9. Her er behovet og kostnadene i forbindelse med datavarehus diskutert.

I implementeringseksempelet i denne oppgaven var det ikke spesielt lønnsomt å implementere et datavarehus i tillegg til databasen. Her tok spørringene uansett bare noen få millisekunder. Det er først når forskjellen i responstid mellom databasene og datavarehuset blir noen sekunder eller mer at man vil merke nytten av å ha et datavarehus. Kravene til responstid er noen få sekunder eller toppen noen få minutter, [7]. Dessuten var ikke datavarehuset betydelig raskere å spørre mot enn databasen. Det skyldes først og fremst at databasen inneholder veldig få nullverdier.

I et tilfelle der databasen er mer komplisert og inneholder flere nullverdier, ville det vært mer lønnsomt med et datavarehus. Da ville forskjellen i responstidene vært betydelig større.

Formålet med datavarehus er typisk at det støtter tidsserie- og trendanalyser. Dette krever mer historiske data enn hva som er vanlig i transaksjonsdatabaser, [4]. Datavarehuset og de underliggende databasene skal altså dekke ulike behov. Ved implementasjon av et datavarehus frigjøres det plass i databasene, og datavarehuset og databasen blir mer effektive hver for seg.

Det er en lang og omfattende prosess å implementere et datavarehus som dekker informasjonen om en hel bedrift, [3]. Noen velger derfor å bruke en datamart istedenfor. En datamart fungerer som et datavarehus og kjører på en datavarehusserver, men den inneholder typisk mye mindre data. Man kan for eksempel ha en datamart som inneholder informasjon om en avdeling i en bedrift.

8.4 Subjektiv vurdering av hovedfagsarbeidet

Arbeidet med denne hovedoppgaven har vært lærerikt og interessant. Det har gitt meg god erfaring med å lese og forstå vitenskapelige artikler. Datavarehus har vært et spennende tema å jobbe med. Jeg har lært at kildefiler ikke alltid er enkle å sette seg inn i. Det var en lang og lærerik prosess å bli kjent med kildefilene og finne en struktur for databasene. Utviklingen i oppgaven har vist meg hvordan problemstillingen i en oppgave kan forandre seg underveis. Det opprinnelige målet med denne oppgaven ble ikke gjennomført, fordi arbeidet med å skaffe plattformen som oppgaven skulle gjøres på ble for komplisert og tidkrevende. Oppgavens innhold måtte endres for å få gjennomført den innen den tidsrammen som er satt for en hovedoppgave.

Referanser

[1] BIBSYSMARC - Bibliografisk format - katalogiseringsregler. http://www.bibsys.no/handbok/marc/marc.pdf.

[2] Surajit Chaudhuri og Umeshwar Dayal. Data warehousing and olap for decision support. *SIGMOD Record*, side 507–508, 1997.

[3] Surajit Chaudhuri og Umeshwar Dayal. An overview of data warehousing and olap technology. *SIGMOD Record*, side 65–74, 1997.

[4] Ramez Elmasri og Shankant B. Navathe. *Fundamentals of Database Systems*. Addison Wesley, 3. utgave, 2000.

[5] Hector Garcia-Molina, Jeffrey D. Ullman og Jennifer Widom. *Database Systems - The Complete Book*. Prentice Hall, 3. utgave, 2002.

[6] Gary Hallmark. The oracle warehouse. *Morgan Kaufmann Publishers Inc.*, side 707–709, 1995.

[7] Venky Harinarayan, Anand Rajaraman og Jeffrey D Ullman. Implementing data cubes efficiently. *ACM Press*, side 205–216, 1996.

[8] Java 2 Platform, Standard Edition, v 1.4.2 API Specification. http://java.sun.com/j2se/1.4.2/docs/api/index.html.

[9] Marcus Jürgens. *Index Structures for Data Warehouses*. Freie Universitet, Berlin, 2003.

[10] Do Lyman, Pamela Drew, Wei Jin, Vish Jumani og David Van Rossum. Issues in developing very large data warehouses. *Morgan Kaufmann Publishers Inc.*, side 633 – 636, 1998.

[11] NORMARC: Format for utveksling av bibliografiske data i maskinleselig form. - 2. utg./revidert av Den norske katalogkomité. - Oslo: Nasjonalbiblioteket, 1999. - 1 b.; 30 cm isbn 82-7965-011-3 (h.). http://www.nb.no/katkom/NORMARC/normarc.html.

[12] Oracle9i database online documentation, release 2 (9.2). http://www.ifi.uio.no/doc/oracle/ora9i/.

[13] Dimitri Theodoratos og Timos Sellis. Data warehouse configuration. *Morgan Kaufmann Publishers Inc.*, side 126 – 135, 1997.

9 Vedleggsoversikt

Vedlegg A Javaprogrammet som leser inn den første relasjonelle databasen, utifra det opprinnelige datagrunnlaget.

Vedlegg B NIAM-analysen av den første relasjonelle databasen.

Vedlegg C Oracleskjema for den første relasjonelle databasen.

Vedlegg D Javaprogrammet som leser inn den andre relasjonelle databasen, utifra det nye datagrunnlaget.

Vedlegg E NIAM-analysen av den andre relasjonelle databasen.

Vedlegg F Oracleskjema for den andre relasjonelle databasen.

A Javaprogram for den første relasjonelle databasen

```
1  import java.sql.*;
2  import oracle.sql.*;
3  import oracle.jdbc.driver.*;
4  import easyIO.*;
5  import java.util.*;
6  import java.io.*;
7
8  public class lese {
9      //Deklarasjoner
10     private In inn;
11     private HashMap bok = null;
12     private Connection conn;
13     private int personid = 0;
14     private int konferanseid = 0;
15     private int korporasjonid = 0;
16     private int tittelid = 0;
17     private int ny_levid = 0;
18
19     public lese(String filnavn) throws SQLException{
20
21         // Last ned Oracle JDBC driveren
22         DriverManager.registerDriver(new
23             oracle.jdbc.driver.OracleDriver());
24
25         //Koble til Oracle
26         kobleTil();
27
28         //Lese dokumenter fra fila
29         inn = new In(filnavn);
30         lesFraFil();
31
32         System.exit(0);
33
34     }//end konstruktør lese
35
36     public void kobleTil() {
37         In tastatur = new In();
38         String brukernavn;
39         String passord;
40
41         System.out.print("Brukernavn: ");
42         brukernavn = tastatur.inString();
43         System.out.print("Passord: ");
44         passord = tastatur.inString();
45
46         try {
47             conn = DriverManager.getConnection(
48 "jdbc:oracle:thin:@blot.ifi.uio.no:1521:IFIFORSK",
49 brukernavn, passord);
50         }
51         catch(Exception e) {
```

```
52              System.err.println("Feil: " + e);
53              System.exit(1);
54          }
55
56
57      }//end kobleTil
58
59      public void lesFraFil() throws SQLException {
60
61          char [] kode = new char[3];
62          char delkode;
63          int helekoden;
64          int count = 0;
65          String tegn;
66
67          tegn = inn.inWord();
68          while(!tegn.startsWith("*")
69                && !tegn.startsWith("^")){
70              tegn = inn.inWord();
71          }
72
73          while(!inn.lastItem()){
74              //Når metodene er ferdig returnerer
75              //de ordet de er på.
76              //Ordet er da ^ eller *xxx.
77
78              //Hvis ordet begynner med ^,
79              //les videre til koden.
80              if(tegn.equals("^")) {
81                  //Da har vi en ny bok
82                  if(count > 0) {
83                      //Ferdig med en bok.
84                      //Den må da legges inn i Oracle
85                      leggInn(bok);
86                      //HashMapen nulles ut
87                      bok = new HashMap();
88
89                      //Nuller ut ArrayListene
90                      ArrayList noter_for = new ArrayList();
91                      bok.put("noter_for", noter_for);
92                      ArrayList emneordene = new ArrayList();
93                      bok.put("emneordene", emneordene);
94                      ArrayList nokkelordene = new ArrayList();
95                      bok.put("nokkelordene", nokkelordene);
96                      ArrayList lokale_ordene = new ArrayList();
97                      bok.put("lokale_ordene", lokale_ordene);
98                      ArrayList urlene = new ArrayList();
99                      bok.put("urlene", urlene);
100                     ArrayList navnene_leverer = new ArrayList();
101                     bok.put("navnene_leverer", navnene_leverer);
102                     ArrayList bibl_kontr_nr = new ArrayList();
103                     bok.put("bibl_kontr_nr", bibl_kontr_nr);
104                     ArrayList riktig_issn = new ArrayList();
105                     bok.put("riktig_issn", riktig_issn);
```

```
106                     ArrayList feil_issn_nr = new ArrayList();
107                     bok.put("feil_issn_nr", feil_issn_nr);
108                     ArrayList fagkoder = new ArrayList();
109                     bok.put("fagkoder", fagkoder);
110                     ArrayList leverandorene = new ArrayList();
111                     bok.put("leverandorene", leverandorene);
112                     ArrayList titlene = new ArrayList();
113                     bok.put("titlene", titlene);
114                     ArrayList personene = new ArrayList();
115                     bok.put("personene", personene);
116                     ArrayList korpene = new ArrayList();
117                     bok.put("korpene", korpene);
118                     ArrayList konfene = new ArrayList();
119                     bok.put("konfene", konfene);
120                     ArrayList isbn_info = new ArrayList();
121                     bok.put("isbn_info", isbn_info);
122
123                 }//end if
124                 else {
125                     //Første boka,
126                     //oppretter da en HashMap bok
127                     bok = new HashMap();
128
129                     //Oppretter ArrayLister der metoden
130                     //kalles flere ganger
131                     ArrayList noter_for = new ArrayList();
132                     bok.put("noter_for",noter_for);
133                     ArrayList emneordene = new ArrayList();
134                     bok.put("emneordene", emneordene);
135                     ArrayList nokkelordene = new ArrayList();
136                     bok.put("nokkelordene", nokkelordene);
137                     ArrayList lokale_ordene = new ArrayList();
138                     bok.put("lokale_ordene", lokale_ordene);
139                     ArrayList urlene = new ArrayList();
140                     bok.put("urlene", urlene);
141                     ArrayList navnene_leverer = new ArrayList();
142                     bok.put("navnene_leverer", navnene_leverer);
143                     ArrayList bibl_kontr_nr = new ArrayList();
144                     bok.put("bibl_kontr_nr", bibl_kontr_nr);
145                     ArrayList riktig_issn = new ArrayList();
146                     bok.put("riktig_issn", riktig_issn);
147                     ArrayList feil_issn_nr = new ArrayList();
148                     bok.put("feil_issn_nr", feil_issn_nr);
149                     ArrayList fagkoder = new ArrayList();
150                     bok.put("fagkoder", fagkoder);
151                     ArrayList leverandorene = new ArrayList();
152                     bok.put("leverandorene", leverandorene);
153                     ArrayList titlene = new ArrayList();
154                     bok.put("titlene", titlene);
155                     ArrayList personene = new ArrayList();
156                     bok.put("personene", personene);
157                     ArrayList korpene = new ArrayList();
158                     bok.put("korpene", korpene);
159                     ArrayList konfene = new ArrayList();
```

```
160                     bok.put("konfene", konfene);
161                     ArrayList isbn_info = new ArrayList();
162                     bok.put("isbn_info", isbn_info);
163
164                 }//end else
165                 count++;
166                 tegn = inn.inWord();
167                 //Lese forbi dokumenter uten id,
168                 //men bare *850
169                 while(!tegn.equals("*000")
170                       && !tegn.equals("*001")) {
171                     tegn = inn.inWord();
172                 }//end while
173
174             }//end if ^
175
176             //Hvis ordet begynner med *, så har vi koden.
177             if(tegn.startsWith("*")){
178                 //Finner ny kode
179                 helekoden = new Integer(tegn.substring(1, 4)).
180                     intValue();
181
182                 switch(helekoden) {
183
184                 case 1: tegn = id();
185                     break;
186                 case 8: tegn = infokoder();
187                     break;
188                 case 12: tegn = fagkode();
189                     break;
190                 case 14: tegn = leverandor();
191                     break;
192                 case 15: tegn = kontrollnr();
193                     break;
194                 case 20: tegn = isbn();
195                     break;
196                 case 22: tegn = issn();
197                     break;
198                 case 41: tegn = spraak();
199                     break;
200                 case 62: tegn = crnr();
201                     break;
202                 case 97: tegn = beholdning();
203                     break;
204                 case 100: tegn = person(helekoden);
205                     break;
206                 case 110: tegn = korporasjon(helekoden);
207                     break;
208                 case 111: tegn = konferanse(helekoden);
209                     break;
210                 case 130: tegn = tittel(helekoden, tegn);
211                     break;
212                 case 210: tegn = forkortet_tittel();
213                     break;
```

```
214                 case 222: tegn = nokkel_tittel();
215                     break;
216                 case 240: tegn = tittel(helekoden, tegn);
217                     break;
218                 case 245: tegn = tittel(helekoden, tegn);
219                     break;
220                 case 246: tegn = tittel(helekoden, tegn);
221                     break;
222                 case 250: tegn = utgave();
223                     break;
224                 case 260: tegn = utgivelse();
225                     break;
226                 case 300: tegn = omfang();
227                     break;
228                 case 310: tegn = periodisitet();
229                     break;
230                 case 500: tegn = note();
231                     break;
232                 case 600: tegn = person(helekoden);
233                     break;
234                 case 610: tegn = korporasjon(helekoden);
235                     break;
236                 case 630: tegn = tittel(helekoden, tegn);
237                     break;
238                 case 650: tegn = emneord();
239                     break;
240                 case 653: tegn = nokkelord();
241                     break;
242                 case 687: tegn = lokale_ord();
243                     break;
244                 case 700: tegn = person(helekoden);
245                     break;
246                 case 710: tegn = korporasjon(helekoden);
247                     break;
248                 case 711: tegn = konferanse(helekoden);
249                     break;
250                 case 730: tegn = tittel(helekoden, tegn);
251                     break;
252                 case 740: tegn = tittel(helekoden, tegn);
253                     break;
254                 case 800: tegn = person(helekoden);
255                     break;
256                 case 830: tegn = tittel(helekoden, tegn);
257                     break;
258                 case 850: tegn = dokeier();
259                     break;
260                 case 856: tegn = url();
261                     break;
262                 default: tegn = videre();
263                     break;
264                 }//end switch
265 
266             }//end if*
267 
```

```
268         }//end while
269
270         //legg inn siste bok
271         leggInn(bok);
272
273         inn.close();
274
275     }//end lesFraFil()
276
277     public void leggInn(HashMap h) throws SQLException{
278
279         System.out.println("*************************");
280         System.out.println("id: " + h.get("id"));
281         System.out.println("*************************");
282
283         // Lag et SQL-setningsobjekt
284         Statement stmt = conn.createStatement();
285
286         //Legger attributter inn i tabellen Dokument i Oracle
287         //*************************************************
288
289         try {
290
291             stmt.executeUpdate("INSERT INTO Dokument(Id_for,
292 Aarstall_slutt_aar, Aarstall_start_aar, Aarstall_utg_aar,
293 CR_hovednr, Tid_reg_dato, Bokstav_katalogisering,
294 L_Navn_utgivelse, Navnet_ansvarlig, Navnet_forkortet_tittel,
295 Navnet_nokkel_tittel, Navnet_utg_forlag, Navnet_utg_sted,
296 Navnet_utg_trykkeri, Navnet_utg_trykksted, Lengde_for,
297 Frekvens_utgivelse, Statusen_for, Tilfoyelse_nforklaring,
298 Utgaven_av) VALUES(" + h.get("id") + ",'" +
299 h.get("aarstall_sluttaar") + "','" + h.get("aarstall_startaar")
300 + "','" + h.get("utg_aar") + "','" + h.get("hovednr") + "','"
301 +  h.get("tid_reg_dato") + "','" + h.get("bokstav_katalogisering")
302 + "','" + h.get("utgivelsesland") + "','" +
303 h.get("navnet_ansvarlig") + "','" + h.get("forkortet_tittel")
304 + "','" + h.get("nokkel_tittel") + "','" + h.get("navnet_forlag")
305 + "','" + h.get("utg_sted") + "','" + h.get("navnet_trykkeri")
306 + "','" + h.get("navnet_trykkested") + "','" + h.get("omfang")
307 + "','" + h.get("frekvens_utgivelse") + "','" + h.get("status")
308 + "','" + h.get("nforklaring") + "','" + h.get("utgaven_av") + "')");
309
310
311         //Legger attributter inn i tabellen Dok_kode i Oracle.
312         //*****************************************************
313
314         ArrayList dk = (ArrayList)h.get("fagkoder");
315         for(int i = 0; i < dk.size(); i++) {
316             stmt.executeUpdate("INSERT INTO Dok_kode
317 (F_Kode_lokal, Id_dok) VALUES(" + "'" + dk.get(i) + "'" + "," +
318 h.get("id") + ")");
319         }//end for
320
321
```

```
322         //Legger attributter i tabellen Leverandor og Dok_Lev.
323         //*************************************************
324
325         ArrayList lev = (ArrayList)h.get("leverandorene");
326
327         for(int i = 0; i < lev.size(); i++) {
328             String[] slev = (String[])lev.get(i);
329
330             //Flere dok har lev_id 99
331             if(slev[0].equals("99")) {
332                 ny_levid++;
333                 slev[0] = Integer.toString(ny_levid);
334             }//end if
335
336             if(slev[0] != null) {
337                 ResultSet rs = stmt.executeQuery("SELECT LevId_for
338 FROM Leverandor WHERE LevId_for = " + "'" + slev[0] + "'");
339
340                 //LevId finnes ikke fra før i Leverandor
341                 if(rs.next() == false) {
342                     stmt.executeUpdate("INSERT INTO Leverandor
343 (LevId_for, Adr_adresse, L_Navn_for, Navnet_lev_navn, Postnr_for)
344 VALUES(" + "'" + slev[0] + "'" + "," + "'" + slev[1] + "'" + ","
345 + "'" + slev[2] + "'" + "," + "'" + slev[3] + "'" + "," + "'" +
346 slev[4] + "'" + ")");
347
348                     stmt.executeUpdate("INSERT INTO Dok_Lev
349 (LevId_lev, Id_dok) VALUES(" + "'" + slev[0] + "'" + "," +
350 h.get("id") + ")");
351
352                 }//end if
353
354                 //LevId finnes fra før i Leverandor
355                 else {
356
357                     ResultSet sett = stmt.executeQuery("SELECT
358 Id_dok FROM Dok_Lev WHERE Id_dok = " + h.get("id"));
359
360                     //Legge inn i Dok_Lev
361                     if(sett.next() == false) {
362                         stmt.executeUpdate("INSERT INTO Dok_Lev(
363 LevId_lev, Id_dok) VALUES(" + "'" + slev[0] + "'" + "," +
364 h.get("id") + ")");
365                     }//end if
366                 }//end else
367             }//end if
368         }//end for
369
370
371         //Legger attributter inn i tabellen Lev_beh i Oracle.
372         //*************************************************
373
374         ArrayList lb = (ArrayList)h.get("navnene_leverer");
375         for(int i = 0; i < lb.size(); i++) {
```

```
376             stmt.executeUpdate("INSERT INTO Lev_beh(Id_dok,
377 Navnet_leverer) VALUES(" + h.get("id") + "," + "'" + lb.get(i)
378 + "'" + ")");
379         }
380 
381 
382         //Legger inn i tabellene Bibl_nr og Bibl_kontr_nr.
383         //**********************************************
384 
385         ArrayList bn = (ArrayList)h.get("bibl_kontr_nr");
386 
387         for(int i = 0; i < bn.size(); i++) {
388             String[] sbn = (String[])bn.get(i);
389             ResultSet res = stmt.executeQuery("SELECT B_nr_for
390 FROM Bibl_nr WHERE B_nr_for = " + "'" + sbn[0] + "'");
391 
392             //Legg inn nr og kilde hvis det ikke finnes fra før
393             if(res.next() == false) {
394                 stmt.executeUpdate("INSERT INTO Bibl_nr(B_nr_for,
395 B_Kilde_nr_kilde) VALUES(" + "'" + sbn[0] + "'" + "," + "'" +
396 sbn[1] + "'" + ")");
397             }//end if
398 
399             stmt.executeUpdate("INSERT INTO Bibl_konr_nr(Id_dok,
400 B_nr_bibl_konr) VALUES(" + h.get("id") + "," + "'" + sbn[0]
401 + "'" + ")");
402 
403         }//end for
404 
405 
406         //Legger attributter inn i tabellen ISBN i Oracle.
407         //**********************************************
408 
409         ArrayList lisbn = (ArrayList)h.get("isbn_info");
410 
411         for(int i = 0; i < lisbn.size(); i++) {
412             String[] sisbn = (String[])lisbn.get(i);
413 
414             //legge inn riktig ISBN
415             if(sisbn[0] != null) {
416 
417                 ResultSet fra_for = stmt.executeQuery("SELECT
418 ISBN_nr_for FROM ISBN WHERE ISBN_nr_for = " + "'" +
419 sisbn[0] + "'");
420 
421                 //ISBNnr finnes ikke fra før
422                 if(fra_for.next() == false) {
423                     stmt.executeUpdate("INSERT INTO ISBN(
424 ISBN_nr_for, Betingelsen_leverings, Id_riktig, Info_innbinding,
425 Tilfoyelse_tilfoyelser) VALUES(" + "'" + sisbn[0] + "'" + ","
426 + "'" + sisbn[1] + "'" + "," + h.get("id") + "," + "'" + sisbn[2]
427 + "'" + "," + "'" + sisbn[3] + "'" + ")");
428                 }//end if
429 
```

```
430                 //ISBNnr finnes fra før
431                 else {
432                     stmt.executeUpdate("UPDATE ISBN SET Id_riktig ="
433 + h.get("id") + "WHERE ISBN_nr_for  =" + "'" + sisbn[0] + "'");
434                 }//end else
435 
436             }//end if
437 
438             //legge inn feil ISBN
439             if(sisbn[4] != null) {
440 
441                 ResultSet ff = stmt.executeQuery("SELECT
442 ISBN_nr_for FROM ISBN WHERE ISBN_nr_for = " + "'" + sisbn[4] + "'");
443 
444                 //ISBNnr finnes ikke fra før
445                 if(ff.next() == false) {
446                     stmt.executeUpdate("INSERT INTO ISBN(ISBN_nr_for,
447 Betingelsen_leverings, Id_feil, Info_innbinding,
448 Tilfoyelse_tilfoyelser) VALUES(" + "'" + sisbn[4] + "'" + "," + "'"
449 + sisbn[1] + "'" + "," +  h.get("id") + "," + "'" + sisbn[2] + "'"
450 + "," + "'" + sisbn[3] + "'" + ")");
451                 }//end if
452 
453                 //ISBNnr finnes fra før
454                 else {
455                     stmt.executeUpdate("UPDATE ISBN SET Id_feil ="
456 + h.get("id") + "WHERE ISBN_nr_for  =" + "'" + sisbn[4] + "'");
457                 }//end else if
458             }//end if
459         }//end for
460 
461 
462         //Legger attributter inn i tabellen ISSN i Oracle.
463         //**************************************************
464 
465         //Legge inn riktig ISSN
466         ArrayList r_issn = (ArrayList)h.get("riktig_issn");
467 
468         if(r_issn != null) {
469             for(int i = 0; i < r_issn.size(); i++) {
470 
471                 ResultSet finnes = stmt.executeQuery("SELECT
472 ISSN_nr_for FROM ISSN WHERE ISSN_nr_for = " + "'" +
473 r_issn.get(i) + "'");
474 
475                 //ISSNnr finnes ikke fra før
476                 if(finnes.next() == false) {
477                     stmt.executeUpdate("INSERT INTO ISSN(
478 ISSN_nr_for, Id_riktig) VALUES(" + "'" + r_issn.get(i)
479 + "'" + "," + h.get("id") + ")");
480                 }//end if
481 
482                 //ISSNnr finnes fra før
483                 else {
```

```
484                     stmt.executeUpdate("UPDATE ISSN SET Id_riktig ="
485 + h.get("id") + "WHERE ISSN_nr_for  =" + "'" + r_issn.get(i) + "'");
486                     }//end else
487                 }//end for
488             }//end if
489 
490             //Legge inn feil ISSN
491             ArrayList f_issn = (ArrayList)h.get("feil_issn_nr");
492 
493             if(f_issn != null) {
494                 for(int i = 0; i < f_issn.size(); i++) {
495 
496                     ResultSet finn = stmt.executeQuery("SELECT
497 ISSN_nr_for FROM ISSN WHERE ISSN_nr_for = " + "'" +
498 f_issn.get(i) + "'");
499 
500                     //ISSNnr finnes ikke fra før
501                     if(finn.next() == false) {
502                         stmt.executeUpdate("INSERT INTO ISSN(
503 ISSN_nr_for, Id_feil) VALUES(" + "'" + f_issn.get(i) + "'"
504 + "," + h.get("id") + ")");
505                     }//end if
506 
507                     //ISSNnr finnes fra før
508                     else {
509                         stmt.executeUpdate("UPDATE ISSN SET Id_feil ="
510 + h.get("id") + "WHERE ISSN_nr_for  =" + "'" + f_issn.get(i) + "'");
511                     }//end else
512                 }//end for
513             }//end if
514 
515             //Legger attributter inn i tabellen Bokspraak i Oracle.
516             //*****************************************************
517 
518             ArrayList sp = (ArrayList)h.get("spraakene");
519             if(sp != null) {
520                 for(int i = 0; i < sp.size(); i++) {
521                     //Må sjekke at ikke samme språk ligger i
522                     //bokspraak og spraakene.
523                     if(!sp.get(i).equals(h.get("bokspraak"))) {
524                         stmt.executeUpdate("INSERT INTO Bokspraak(
525 Id_skrevet_paa, Spraak_brukt_i) VALUES(" + h.get("id") + "," + "'"
526 + sp.get(i) + "'" + ")");
527                     }//end if
528                 }//end for
529             }//end if
530 
531             if(h.get("bokspraak") != null) {
532                 stmt.executeUpdate("INSERT INTO Bokspraak(Id_skrevet_paa,
533 Spraak_brukt_i) VALUES(" + h.get("id") + "," + "'" +
534 h.get("bokspraak") + "'" + ")");
535             }//end if
536 
537 
```

```
538         //Legger attributter inn i tabellen AltNr i Oracle.
539         //*************************************************
540
541         ArrayList an = (ArrayList)h.get("altnr");
542
543         if(an != null) {
544             for(int i = 0; i < an.size(); i++) {
545                 stmt.executeUpdate("INSERT INTO AltNr(Id_har,
546 CR_alt_nr) VALUES(" + h.get("id") + "," + "'" + an.get(i)
547 + "'" + ")");
548             }//end for
549         }//end if
550
551
552         //Legger inn i tabellene Person og Ans_Pers i Oracle.
553         //*************************************************
554
555         ArrayList lpers = (ArrayList)h.get("personene");
556
557         for(int i = 0; i < lpers.size(); i++) {
558             String[] spers = (String[])lpers.get(i);
559
560             if(spers[1] != null) {
561             stmt.executeUpdate("INSERT INTO Person(P_Id_for,
562 Navnet_paa, Aarstall_for, L_Navn_nasjonalitet,
563 Tilfoyelse_tilfoyelser) VALUES(" + "'" + spers[1] + "'" + ","
564 + "'" + spers[2] + "'" + "," + "'" + spers[3] + "'" + "," + "'"
565 + spers[4] + "'" + "," + "'" + spers[5] + "'" + ")");
566
567             stmt.executeUpdate("INSERT INTO Ans_Pers(Id_bok, P_Id_er,
568 Koden_for) VALUES(" + h.get("id") + "," + "'" + spers[1] + "'" + ","
569 + "'" + spers[0] + "'" + ")");
570
571             }//end if
572         }//end for
573
574
575         //Legger inn i tabellene Konferanse og Ans_Konf i Oracle.
576         //*************************************************
577
578         ArrayList lkonf = (ArrayList)h.get("konfene");
579
580         for(int i = 0; i < lkonf.size(); i++) {
581             String[] skonf = (String[])lkonf.get(i);
582
583             if(skonf[1] != null) {
584                 stmt.executeUpdate("INSERT INTO Konferanse(
585 Konf_id_for, Tid_for, Navnet_konf_navn, Navnet_stedsnavn, Tall_paa)
586 VALUES(" + "'" + skonf[1] + "'" + "," + "'" + skonf[2] + "'"
587 + "," + "'" + skonf[3] + "'" + "," + "'" + skonf[4] + "'" + "," + "'"
588 + skonf[5] + "'" + ")");
589
590             stmt.executeUpdate("INSERT INTO Ans_Konf(Id_dok,
591 Konf_id_ansvarlig, Koden_for) VALUES(" + h.get("id") + "," + "'" +
```

```
592 skonf[1] + "'" + "," + "'" + skonf[0] + "'" + ")");
593             }//end if
594 
595         }//end for
596 
597 
598         //Legger inn i tabellene Korporasjon og Ans_Korp i Oracle.
599         //*****************************************************
600 
601         ArrayList lkorp = (ArrayList)h.get("korpene");
602 
603         for(int i = 0; i < lkorp.size(); i++) {
604             String[] skorp = (String[])lkorp.get(i);
605 
606             if(!skorp[1].equals(null)) {
607             stmt.executeUpdate("INSERT INTO Korporasjon(Korp_id_for,
608 Tid_for, Navnet_korp_navn, Navnet_stedsnavn, Tall_for) VALUES(" + "'"
609 + skorp[1] + "'" + "," + "'" + skorp[2] + "'" + "," + "'" + skorp[3]
610 + "'" + "," + "'" + skorp[4] + "'" + "," + "'" + skorp[5] + "'"
611 + ")");
612 
613             stmt.executeUpdate("INSERT INTO Ans_Korp(Id_dok,
614 Korp_id_ansvarlig, Koden_for) VALUES(" + h.get("id") + "," + "'" +
615 skorp[1] + "'" + "," + "'" + skorp[0] + "'" + ")");
616 
617             }//end if
618         }//end for
619 
620 
621         //Legger attributter inn i tabellen Tittel i Oracle.
622         //*************************************************
623 
624         ArrayList ltitt = (ArrayList)h.get("titlene");
625 
626         for(int i = 0; i < ltitt.size(); i++) {
627             String[] stitt = (String[])ltitt.get(i);
628 
629             stmt.executeUpdate("INSERT INTO Tittel(T_id_for,
630 Id_dok, Koden_t_kode, Navnet_hovedtittel, Aarstall_aarstall,
631 Aarstall_utgivelsesaar, Info_om) VALUES(" + "'" + stitt[1] + "'"
632 + "," + h.get("id") + "," + "'" + stitt[0] + "'" + "," + "'" +
633 stitt[2] + "'" + "," + "'" + stitt[3] + "'" + "," + "'" +
634 stitt[4] + "'" + "," + "'" + stitt[5] + "'" + ")");
635 
636         }//end for
637 
638 
639         //Legger attributter inn i tabellen Dok_Note i Oracle.
640         //*************************************************
641 
642         ArrayList dn = (ArrayList)h.get("noter_for");
643         for(int i = 0; i < dn.size(); i++) {
644             stmt.executeUpdate("INSERT INTO Dok_Note(Id_dok,
645 Noten_note) VALUES(" + h.get("id") + "," + "'" + dn.get(i)
```

```
646 + "'" + ")");
647         }//end for
648 
649 
650         //Legger attributter inn i tabellen Emneord i Oracle.
651         //*************************************************
652 
653         ArrayList eo = (ArrayList)h.get("emneordene");
654         for(int i = 0; i < eo.size(); i++) {
655             stmt.executeUpdate("INSERT INTO Emneord(Id_bok,
656 Ordet_ord) VALUES(" + h.get("id") + "," + "'" + eo.get(i)
657 + "'" + ")");
658         }//end for
659 
660 
661         //Legger attributter inn i tabellen Nokkelord i Oracle.
662         //*****************************************************
663 
664         ArrayList no = (ArrayList)h.get("nokkelordene");
665         for(int i = 0; i < no.size(); i++) {
666             stmt.executeUpdate("INSERT INTO Nokkelord(Id_bok,
667 Ordet_ord) VALUES(" + h.get("id") + "," + "'" + no.get(i)
668 + "'" + ")");
669         }//end for
670 
671 
672         //Legger attributter inn i tabellen Lokale_Ord i Oracle.
673         //*******************************************************
674 
675         ArrayList lo = (ArrayList)h.get("lokale_ordene");
676         for(int i = 0; i < lo.size(); i++) {
677             stmt.executeUpdate("INSERT INTO Lokale_Ord(Id_bok,
678 Ordet_ord) VALUES(" + h.get("id") + "," + "'" + lo.get(i)
679 + "'" + ")");
680         }//end for
681 
682 
683         //Legger attributter inn i tabellen Dok_Eier i Oracle.
684         //*****************************************************
685 
686         if(!h.get("eier_bibl").equals(null)) {
687             stmt.executeUpdate("INSERT INTO Dok_Eier(Id_har,
688 Navnet_eier, Ant_eks, Filial_for, Hylle_plass) VALUES(" +
689 h.get("id") + "," + "'" + h.get("eier_bibl") + "'" + "," + "'" +
690 h.get("ant_eks") + "'" + "," + "'" + h.get("filial") + "'" + ","
691 + "'" + h.get("hylle") + "'" + ")");
692         }//end if
693 
694         //Legger attributter i tabellen URL og URL_Dok i Oracle.
695         //*****************************************************
696 
697         ArrayList urler = (ArrayList)h.get("urlene");
698         for(int i = 0; i < urler.size(); i++) {
699             //Sjekker om URL finnes fra før
```

```
700             ResultSet finn_url = stmt.executeQuery("SELECT
701 URL_Adr_for FROM URL WHERE URL_Adr_for = " + "'" + urler.get(i)
702 + "'");
703             if(finn_url.next() == false) {
704                 stmt.executeUpdate("INSERT INTO URL(URL_Adr_for,
705 Storrelse_str, Formatet_for, Noten_ekstern, Noten_intern) VALUES("
706 + "'" + urler.get(i) + "'" + "," + "'" + h.get("filstorrelse") + "'"
707 + "," + "'" + h.get("format") + "'" + "," + "'" + h.get("note") + "'"
708 + "," + "'" + h.get("intern_note") + "'" + ")");
709             }//end if
710
711             stmt.executeUpdate("INSERT INTO URL_Dok(Id_dok,
712 URL_Adr_url) VALUES(" + h.get("id") + "," + "'" + urler.get(i)
713 + "'" + ")");
714
715         }//end for
716
717         stmt.close();
718
719         }catch(Exception e) {
720             System.err.println(e);
721             System.exit(1);
722         }
723
724     }//end leggInn()
725
726
727     public String videre() {
728         //Lese videre til neste kode
729         String vid = null;
730
731         vid = inn.inWord();
732         while(!vid.startsWith("*") && !vid.startsWith("^")) {
733             vid = inn.inWord();
734         }//end while
735
736         return vid;
737
738     }//end videre()
739
740
741     public String id() {
742         //Lese dokumentId fra fila og legge den i HashMap bok
743         String id = null;
744         String nest = null;
745
746         id = inn.inWord();
747         bok.put("id",id);
748
749         nest = inn.inWord();
750         while(!nest.startsWith("*") && !nest.startsWith("^")) {
751             nest = inn.inWord();
752         }//end while
753
```

```
754         return nest;
755
756     }//end id()
757
758
759     public String infokoder() {
760         //Lese informasjonskoder fra fila og legge inn i bok
761         //$g og 11-14 er aarstall_sluttaar, $i og 18 er frek_utg,
762         //00-05 er tid_reg_dato, 35-37 og $c er språk,
763         //15-17 og $h er utgivelsesland, 39 er bokstav_kat,
764         //$f og 07-10 er aarstall_startaar og $e og 06 er status.
765
766         String aarstall_sluttaar = null;
767         String aarstall_startaar = null;
768         String tid_reg_dato = null;
769         String bokstav_katalogisering = null;
770         String utgivelsesland = null;
771         String frekvens_utgivelse = null;
772         String status = null;
773         String bokspraak= null;
774         String enn = null;
775         String ennenn = null;
776         String linja = null;
777         int lengde = 0;
778         int igjen = 0;
779         String nest = null;
780
781         enn = inn.inWord();
782         while(enn.startsWith("$")) {
783             if(enn.startsWith("$e")) {
784                 //Status er en bokstav
785                 status = enn.substring(2);
786                 ennenn = inn.inWord();
787                 while(!ennenn.startsWith("$")
788                       && !ennenn.startsWith("*")
789                       && !ennenn.startsWith("^")) {
790                     status = status + ennenn;
791                     ennenn = inn.inWord();
792                 }
793                 //Legger inn i HashMap bok
794                 bok.put("status",status);
795                 enn = ennenn;
796             }//end $e
797             else if(enn.startsWith("$f")) {
798                 //Startår er fire sifre, ukjent siffer = u
799                 aarstall_startaar = enn.substring(2);
800                 ennenn = inn.inWord();
801                 while(!ennenn.startsWith("$")
802                       && !ennenn.startsWith("*")
803                       && !ennenn.startsWith("^")) {
804                     aarstall_startaar = aarstall_startaar + ennenn;
805                     ennenn = inn.inWord();
806                 }
807                 //Legger inn i HashMap bok
```

```
808                 bok.put("aarstall_startaar",aarstall_startaar);
809                 enn = ennenn;
810             }//end $f
811             else if(enn.startsWith("$g")) {
812                 //Sluttår er fire sifre, ukjent siffer = u
813                 aarstall_sluttaar = enn.substring(2);
814                 ennenn = inn.inWord();
815                 while(!ennenn.startsWith("$")
816                       && !ennenn.startsWith("*")
817                       && !ennenn.startsWith("^")) {
818                     aarstall_sluttaar = aarstall_sluttaar + ennenn;
819                     ennenn = inn.inWord();
820                 }
821                 //Legger inn i HashMap bok
822                 bok.put("aarstall_sluttaar",aarstall_sluttaar);
823                 enn = ennenn;
824             }//end $g
825             else if(enn.startsWith("$i")) {
826                 //Frekvens er en bokstav
827                 frekvens_utgivelse = enn.substring(2);
828                 ennenn = inn.inWord();
829                 while(!ennenn.startsWith("$")
830                       && !ennenn.startsWith("*")
831                       && !ennenn.startsWith("^")) {
832                     frekvens_utgivelse = frekvens_utgivelse + ennenn;
833                     ennenn = inn.inWord();
834                 }
835                 //Legger inn i HashMap bok
836                 bok.put("frekvens_utgivelse",frekvens_utgivelse);
837                 enn = ennenn;
838             }//end $i
839             else {
840                 enn = inn.inWord();
841                 while(!enn.startsWith("$") && !enn.startsWith("*") &&
842                       !enn.startsWith("^")) {
843                     enn = inn.inWord();
844                 }//end while
845             }//end else
846
847         }//end while
848
849
850         //Ikke $, men streng av lengde 40
851         if(!enn.startsWith("*") && !enn.startsWith("$") &&
852            !enn.startsWith("^")) {
853         lengde = enn.length();
854         igjen = 40 - lengde;
855         linja = enn + inn.inChar();
856         for(int i = 0; i < igjen; i++) {
857             linja = linja + inn.inChar();
858                 }
859
860         tid_reg_dato = linja.substring(0,6);
861         if(tid_reg_dato.equals("      ")) {
```

```
862             tid_reg_dato = null;
863         }
864         bok.put("tid_reg_dato", tid_reg_dato);
865 
866         status = linja.substring(6,7);
867         if(status.equals(" ")) {
868             status = null;
869         }
870         bok.put("status", status);
871 
872         aarstall_startaar = linja.substring(7,11);
873         if(aarstall_startaar.equals("    ")) {
874             aarstall_startaar = null;
875         }
876         bok.put("aarstall_startaar", aarstall_startaar);
877 
878         aarstall_sluttaar = linja.substring(11,15);
879         if(aarstall_sluttaar.equals("    ")) {
880             aarstall_sluttaar = null;
881         }
882         bok.put("aarstall_sluttaar", aarstall_sluttaar);
883 
884         utgivelsesland = linja.substring(15,18);
885         if(utgivelsesland.equals("   ")) {
886             utgivelsesland = null;
887         }
888         bok.put("utgivelsesland", utgivelsesland);
889 
890         bokspraak = linja.substring(35,38);
891         if(!bokspraak.equals("mul") && !bokspraak.equals("   ")) {
892         bok.put("bokspraak", bokspraak);
893         }
894 
895         bokstav_katalogisering = linja.substring(39,40);
896         if(bokstav_katalogisering.equals(" ")) {
897             bokstav_katalogisering = null;
898         }
899         bok.put("bokstav_katalogisering", bokstav_katalogisering);
900 
901         frekvens_utgivelse = linja.substring(18,19);
902         if(frekvens_utgivelse.equals(" ")) {
903             frekvens_utgivelse = null;
904         }
905         bok.put("frekvens_utgivelse", frekvens_utgivelse);
906 
907         enn = inn.inWord();
908         while(!enn.startsWith("*") && !enn.startsWith("^")) {
909             enn = inn.inWord();
910         }//end while
911 
912         }//end if
913 
914         return enn;
915     }//end infokoder
```

```
916
917
918     public String fagkode() {
919
920         String fagkode = null;
921         String nest = null;
922
923         inn.inWord();//Hopper over $a
924         fagkode = inn.inWord();//Leser inn fagkoden
925
926         ArrayList l = (ArrayList)bok.get("fagkoder");
927
928         //Må sjekke om fagkoden fins fra før.
929         if(!l.contains(fagkode)) {
930             l.add(fagkode);
931         }
932
933         nest = inn.inWord();
934         while(!nest.startsWith("*") && !nest.startsWith("^")) {
935             nest = inn.inWord();
936         }//end while
937
938         return nest;
939
940     }//end fagkode()
941
942
943     public String leverandor() {
944
945         //Lese id, navn, adresse, poststed/nr, land for leverandør
946         //fra fila og legge det i HashMap bok
947         String naa = null;
948         String etterpaa = null;
949         String lev_id = null;
950         String lev_navn = null;
951         String lev_adr = null;
952         String lev_post = null;
953         String lev_land = null;
954         String leverandor[] = new String[5];
955
956         naa = inn.inWord();
957         while(naa.startsWith("$")) {
958             if(naa.startsWith("$a")) {
959                 lev_id = naa.substring(2);
960                 etterpaa = inn.inWord();
961                 while(!etterpaa.startsWith("$")
962                       && !etterpaa.startsWith("*")
963                       && !etterpaa.startsWith("^")) {
964                     lev_id = lev_id + " " + etterpaa;
965                     etterpaa = inn.inWord();
966                 }
967                 lev_id = lev_id.trim();
968                 leverandor[0] = lev_id;
969                 naa = etterpaa;
```

```
970             }//end $a
971             else if(naa.startsWith("$b")) {
972                 lev_navn = naa.substring(2);
973                 etterpaa = inn.inWord();
974                 while(!etterpaa.startsWith("$")
975                       && !etterpaa.startsWith("*")
976                       && !etterpaa.startsWith("^")) {
977                     lev_navn = lev_navn + " " + etterpaa;
978                     etterpaa = inn.inWord();
979                 }
980                 lev_navn = lev_navn.trim();
981                 leverandor[3] = lev_navn;
982                 naa = etterpaa;
983             }//end $b
984             else if(naa.startsWith("$c")) {
985                 lev_adr = naa.substring(2);
986                 etterpaa = inn.inWord();
987                 while(!etterpaa.startsWith("$")
988                       && !etterpaa.startsWith("*")
989                       && !etterpaa.startsWith("^")) {
990                     lev_adr = lev_adr + " " + etterpaa;
991                     etterpaa = inn.inWord();
992                 }
993                 lev_adr = lev_adr.trim();
994                 leverandor[1] = lev_adr;
995                 naa = etterpaa;
996             }//end $c
997             else if(naa.startsWith("$d")) {
998                 lev_post = naa.substring(2);
999                 etterpaa = inn.inWord();
1000                while(!etterpaa.startsWith("$")
1001                      && !etterpaa.startsWith("*")
1002                      && !etterpaa.startsWith("^")) {
1003                    lev_post = lev_post + " " + etterpaa;
1004                    etterpaa = inn.inWord();
1005                }
1006                lev_post = lev_post.trim();
1007                leverandor[4] = lev_post;
1008                naa = etterpaa;
1009            }//end $d
1010            else if(naa.startsWith("$e")) {
1011                lev_land = naa.substring(2);
1012                etterpaa = inn.inWord();
1013                while(!etterpaa.startsWith("$")
1014                      && !etterpaa.startsWith("*")
1015                      && !etterpaa.startsWith("^")) {
1016                    lev_land = lev_land + " " + etterpaa;
1017                    etterpaa = inn.inWord();
1018                }
1019                lev_land = lev_land.trim();
1020                leverandor[2] = lev_land;
1021                naa = etterpaa;
1022            }//end $e
1023            else {
```

```
1024                naa = inn.inWord();
1025                while(!naa.startsWith("$") && !naa.startsWith("*")
1026                      && !naa.startsWith("^")) {
1027                    naa = inn.inWord();
1028                }//end while
1029            }//end else
1030        }//end while
1031
1032        ArrayList l = (ArrayList)bok.get("leverandorene");
1033        l.add(leverandor);
1034
1035        return naa;
1036
1037    }//end leverandor()
1038
1039
1040    public String kontrollnr() {
1041        //lese bibliografisk kontrollnr og kilde og legge inn i bok
1042
1043        String tall_bibl_kontr = null;
1044        String kilde_kontrollnr = null;
1045        String and = null;
1046        String andand = null;
1047        String bibl_nr_kilde[] = new String[2];
1048
1049        and = inn.inWord();
1050        while(and.startsWith("$")) {
1051            if(and.startsWith("$a")) {
1052                tall_bibl_kontr = and.substring(2);
1053                andand = inn.inWord();
1054                while(!andand.startsWith("$")
1055                      && !andand.startsWith("*")
1056                      && !andand.startsWith("^")) {
1057                    tall_bibl_kontr = tall_bibl_kontr + andand;
1058                    andand = inn.inWord();
1059                }
1060                bibl_nr_kilde[0] = tall_bibl_kontr;
1061                and = andand;
1062            }//end $a
1063            else if(and.startsWith("$b")) {
1064                kilde_kontrollnr = and.substring(2);
1065                andand = inn.inWord();
1066                while(!andand.startsWith("$")
1067                      && !andand.startsWith("*")
1068                      && !andand.startsWith("^")) {
1069                    kilde_kontrollnr = kilde_kontrollnr
1070                        + " " + andand;
1071                    andand = inn.inWord();
1072                }
1073                bibl_nr_kilde[1] = kilde_kontrollnr;
1074                and = andand;
1075            }//end $b
1076            else {
1077                and = inn.inWord();
```

```
1078                while(!and.startsWith("$") && !and.startsWith("*")
1079                      && !and.startsWith("^")) {
1080                    and = inn.inWord();
1081                }//end while
1082            }//end else
1083        }//end while
1084
1085        ArrayList l = (ArrayList)bok.get("bibl_kontr_nr");
1086        l.add(bibl_nr_kilde);
1087
1088        return and;
1089
1090    }//end kontrollnr()
1091
1092
1093    public String isbn() {
1094        //Lese ISBN, feil_ISBN, leveringsbetingelse, innbindingsinfo
1095        //og andre tilføyelser og legge det inn i bok.
1096        String denne = null;
1097        String annen = null;
1098        String riktig_isbn = null;
1099        String feil_isbn = null;
1100        String info_innbinding = null;
1101        String betingelsen_leverings = null;
1102        String tilfoyelse_tilfoyelser = null;
1103        String isbn_i[] = new String[5];
1104
1105        denne = inn.inWord();
1106        while(denne.startsWith("$")) {
1107            if(denne.startsWith("$a")) {
1108                riktig_isbn = denne.substring(2);
1109                annen = inn.inWord();
1110                while(!annen.startsWith("$")
1111                      && !annen.startsWith("*")
1112                      && !annen.startsWith("^")) {
1113                    riktig_isbn = riktig_isbn + annen;
1114                    annen = inn.inWord();
1115                }
1116                isbn_i[0] = riktig_isbn;
1117                denne = annen;
1118            }//end $a
1119            else if(denne.startsWith("$b")) {
1120                info_innbinding = denne.substring(2);
1121                annen = inn.inWord();
1122                while(!annen.startsWith("$")
1123                      && !annen.startsWith("*")
1124                      && !annen.startsWith("^")) {
1125                    info_innbinding = info_innbinding + " " + annen;
1126                    annen = inn.inWord();
1127                }
1128                isbn_i[2] = info_innbinding;
1129                denne = annen;
1130            }//end $b
1131            else if(denne.startsWith("$c")) {
```

```
1132                betingelsen_leverings = denne.substring(2);
1133                annen = inn.inWord();
1134                while(!annen.startsWith("$")
1135                      && !annen.startsWith("*")
1136                      && !annen.startsWith("^")) {
1137                    betingelsen_leverings =
1138                        betingelsen_leverings +" "+ annen;
1139                    annen = inn.inWord();
1140                }
1141                isbn_i[1] = betingelsen_leverings;
1142                denne = annen;
1143            }//end $c
1144            else if(denne.startsWith("$g")) {
1145                tilfoyelse_tilfoyelser = denne.substring(2);
1146                annen = inn.inWord();
1147                while(!annen.startsWith("$")
1148                      && !annen.startsWith("*")
1149                      && !annen.startsWith("^")) {
1150                    tilfoyelse_tilfoyelser =
1151                        tilfoyelse_tilfoyelser+" "+annen;
1152                    annen = inn.inWord();
1153                }
1154                isbn_i[3] = tilfoyelse_tilfoyelser;
1155                denne = annen;
1156            }//end $g
1157            else if(denne.startsWith("$z")) {
1158                feil_isbn = denne.substring(2);
1159                annen = inn.inWord();
1160                while(!annen.startsWith("$")
1161                      && !annen.startsWith("*")
1162                      && !annen.startsWith("^")) {
1163                    feil_isbn = feil_isbn + " " + annen;
1164                    annen = inn.inWord();
1165                }
1166                isbn_i[4] = feil_isbn;
1167                denne = annen;
1168            }//end $c
1169            else {
1170                denne = inn.inWord();
1171                while(!denne.startsWith("$")
1172                      && !denne.startsWith("*")
1173                      && !denne.startsWith("^")) {
1174                    denne = inn.inWord();
1175                }//end while
1176            }//end else
1177        }//end while
1178
1179        ArrayList l = (ArrayList)bok.get("isbn_info");
1180        l.add(isbn_i);
1181
1182        return denne;
1183
1184    }//end isbn()
1185
```

```
1186
1187    public String issn() {
1188
1189        //Lese ISSN og feil_ISSN og legge inn i bok.
1190        String den = null;
1191        String and = null;
1192        String issn = null;
1193        String feil_issn = null;
1194
1195        den = inn.inWord();
1196        while(den.startsWith("$")) {
1197            if(den.startsWith("$a")) {
1198                issn = den.substring(2);
1199                and = inn.inWord();
1200                while(!and.startsWith("$") && !and.startsWith("*")
1201                      && !and.startsWith("^")) {
1202                    issn = issn + " " + and;
1203                    and = inn.inWord();
1204                }
1205                ArrayList l = (ArrayList)bok.get("riktig_issn");
1206                l.add(issn);
1207
1208                den = and;
1209            }//end $a
1210            else if(den.startsWith("$y")) {
1211                feil_issn = den.substring(2);
1212                and = inn.inWord();
1213                while(!and.startsWith("$") && !and.startsWith("*")
1214                      && !and.startsWith("^")) {
1215                    feil_issn = feil_issn + " " + and;
1216                    and = inn.inWord();
1217                }
1218                ArrayList l = (ArrayList)bok.get("feil_issn_nr");
1219                l.add(feil_issn);
1220
1221                den = and;
1222            }//end $y
1223            else {
1224                den = inn.inWord();
1225                while(!den.startsWith("$") && !den.startsWith("*")
1226                      && !and.startsWith("^")) {
1227                    den = inn.inWord();
1228                }//end while
1229            }//end else
1230        }//end while
1231
1232        return den;
1233
1234    }//end issn()
1235
1236
1237    public String spraak() {
1238        //Lese inn språk og legge det i bok
1239
```

```
1240        String spraak = null;
1241        String og = null;
1242        String spraaket = null;
1243        ArrayList spraakene = new ArrayList();
1244
1245        og = inn.inWord();
1246        while(og.startsWith("$")) {
1247            if(og.startsWith("$a")) {
1248                spraak = og.substring(2);
1249                for(int i = 0; i < 30; i+=3) {
1250                    if(i+3 <= spraak.length()) {
1251                        spraaket = spraak.substring(i,i+3);
1252                        if(!spraakene.contains(spraaket)) {
1253                            spraakene.add(spraaket);
1254                        }//end if
1255                    }//end if
1256                }//end for
1257                og = inn.inWord();
1258            }//end $a
1259            else {
1260                og = inn.inWord();
1261                while(!og.startsWith("$") && !og.startsWith("*")
1262                      && !og.startsWith("^")) {
1263                    og = inn.inWord();
1264                }//end while
1265            }//end else
1266        }//end while
1267        bok.put("spraakene", spraakene);
1268        return og;
1269    }//end spraak()
1270
1271
1272    public String crnr() {
1273        //Lese CRnr(hovednr og flere altnr) og legg det inn i bok
1274
1275        String dollar = null;
1276        String hovednr = null;
1277        String alt_nr = null;
1278        ArrayList altnr = new ArrayList();
1279        String resten = null;
1280
1281        dollar = inn.inWord();
1282        while(dollar.startsWith("$")) {
1283            //$a er hovednr og $b er altnr
1284            if(dollar.startsWith("$a")) {
1285                hovednr = dollar.substring(2);//tar bort $a
1286                resten = inn.inWord();
1287                while(!resten.startsWith("$")
1288                      && !resten.startsWith("*")
1289                      && !resten.startsWith("^")) {
1290                    hovednr = hovednr + " " + resten;
1291                    resten = inn.inWord();
1292                }//end while
1293                bok.put("hovednr",hovednr);
```

```
1294                dollar = resten;
1295            }//end $a
1296            else if(dollar.startsWith("$b")) {
1297                alt_nr = dollar.substring(2);
1298                resten = inn.inWord();
1299                while(!resten.startsWith("$")
1300                      && !resten.startsWith("*")
1301                      && !resten.startsWith("^")) {
1302                    alt_nr = alt_nr + resten;
1303                    resten = inn.inWord();
1304                }//end while
1305                if(!altnr.contains(alt_nr)) {
1306                    altnr.add(alt_nr);
1307                }
1308                dollar = resten;
1309            }//end $b
1310            else {
1311                dollar = inn.inWord();
1312                while(!dollar.startsWith("$")
1313                      && !dollar.startsWith("*")
1314                      && !dollar.startsWith("^")) {
1315                    dollar = inn.inWord();
1316                }//end while
1317            }//end else
1318
1319        }//end while
1320
1321        bok.put("altnr",altnr);
1322        return dollar;
1323    }//end crnr()
1324
1325
1326    public String beholdning() {
1327        //Lese inn navnet_leverer og legge inn i bok
1328
1329        String beh_ant = null;
1330        String navnet_leverer = null;
1331        String nu = null;
1332        String mernu = null;
1333
1334        nu = inn.inWord();
1335        while(nu.startsWith("$")) {
1336            if(nu.startsWith("$b")) {
1337                navnet_leverer = nu.substring(2);
1338                mernu = inn.inWord();
1339                while(!mernu.startsWith("$")
1340                      && !mernu.startsWith("*")
1341                      && !mernu.startsWith("^")) {
1342                    navnet_leverer = navnet_leverer + " " + mernu;
1343                    mernu = inn.inWord();
1344                }
1345
1346                ArrayList l = (ArrayList)bok.get("navnene_leverer");
1347                l.add(navnet_leverer);
```

```
1348
1349                nu = mernu;
1350            }//end $b
1351            else {
1352                nu = inn.inWord();
1353                while(!nu.startsWith("$") && !nu.startsWith("*")
1354                      && !nu.startsWith("^")) {
1355                    nu = inn.inWord();
1356                }//end while
1357            }//end else
1358        }//end while
1359
1360        return nu;
1361
1362    }//end beholdning()
1363
1364
1365    public String person(int p_kode) {
1366        //Lese navn, tid og tilfoyelse for person og legge inn i bok
1367
1368        String pers_her = null;
1369        String pers_mer = null;
1370        String pers_id = null;
1371        String pers_navn = null;
1372        String pers_tid = null;
1373        String pers_tilfoyelse = null;
1374        String pers_nasjon = null;
1375        String pers_kode = Integer.toString(p_kode);
1376        String pers[] = new String[6];
1377
1378        pers[0] = pers_kode;
1379
1380        personid++;
1381        pers[1] = Integer.toString(personid);
1382
1383        pers_her = inn.inWord();
1384        while(pers_her.startsWith("$") ||
1385              pers_her.charAt(1) == '$') {
1386            if(pers_her.startsWith("$a")) {
1387                pers_navn = pers_her.substring(2);
1388                pers_mer = inn.inWord();
1389                while(!pers_mer.startsWith("$")
1390                      && !pers_mer.startsWith("*")
1391                      && !pers_mer.startsWith("^")) {
1392                    pers_navn = pers_navn + " " + pers_mer;
1393                    pers_mer = inn.inWord();
1394                }
1395                pers[2] = pers_navn;
1396                pers_her = pers_mer;
1397            }//end $a
1398            else if(pers_her.charAt(1) == '$') {
1399                pers_navn = pers_her.substring(3);
1400                pers_mer = inn.inWord();
1401                while(!pers_mer.startsWith("$")
```

```
1402                      && !pers_mer.startsWith("*")
1403                      && !pers_mer.startsWith("^")) {
1404                  pers_navn = pers_navn + " " + pers_mer;
1405                  pers_mer = inn.inWord();
1406              }
1407              pers[2] = pers_navn;
1408              pers_her = pers_mer;
1409          }//end 0$a
1410          else if(pers_her.startsWith("$c")) {
1411              pers_tilfoyelse = pers_her.substring(2);
1412              pers_mer = inn.inWord();
1413              while(!pers_mer.startsWith("$")
1414                      && !pers_mer.startsWith("*")
1415                      && !pers_mer.startsWith("^")) {
1416                  pers_tilfoyelse = pers_tilfoyelse
1417                      + " " + pers_mer;
1418                  pers_mer = inn.inWord();
1419              }
1420              pers[5] = pers_tilfoyelse;
1421              pers_her = pers_mer;
1422          }//end $c
1423          else if(pers_her.startsWith("$d")) {
1424              pers_tid = pers_her.substring(2);
1425              pers_mer = inn.inWord();
1426              while(!pers_mer.startsWith("$")
1427                      && !pers_mer.startsWith("*")
1428                      && !pers_mer.startsWith("^")) {
1429                  pers_tid = pers_tid + " " + pers_mer;
1430                  pers_mer = inn.inWord();
1431              }
1432              pers[3] = pers_tid;
1433              pers_her = pers_mer;
1434          }//end $d
1435          else if(pers_her.startsWith("$j")) {
1436              pers_nasjon = pers_her.substring(2);
1437              pers_mer = inn.inWord();
1438              while(!pers_mer.startsWith("$")
1439                      && !pers_mer.startsWith("*")
1440                      && !pers_mer.startsWith("^")) {
1441                  pers_nasjon = pers_nasjon + " " + pers_mer;
1442                  pers_mer = inn.inWord();
1443              }
1444              pers[4] = pers_nasjon;
1445              pers_her = pers_mer;
1446          }//end $j
1447          else {
1448              pers_her = inn.inWord();
1449              while(!pers_her.startsWith("$")
1450                      && !pers_her.startsWith("*")
1451                      && !pers_her.startsWith("^")) {
1452                  pers_her = inn.inWord();
1453              }//end while
1454          }//end else
1455      }//end while
```

```
1456
1457            ArrayList l = (ArrayList)bok.get("personene");
1458            l.add(pers);
1459
1460            return pers_her;
1461        }//end person()
1462
1463
1464        public String konferanse(int konf_kode) {
1465            //Lese inn navn, sted, dato og nr for konferanse
1466
1467            String her = null;
1468            String mer = null;
1469            String konf_id = null;
1470            String konf_navn = null;
1471            String konf_sted = null;
1472            String konf_nummer = null;
1473            String konf_tid = null;
1474            String konferanse_kode = Integer.toString(konf_kode);
1475            String konf[] = new String[6];
1476
1477            konf[0] = konferanse_kode;
1478
1479            konferanseid++;
1480            konf[1] = Integer.toString(konferanseid);
1481
1482            her = inn.inWord();
1483            while(her.startsWith("$")) {
1484                if(her.startsWith("$a")) {
1485                    konf_navn = her.substring(2);
1486                    mer = inn.inWord();
1487                    while(!mer.startsWith("$") && !mer.startsWith("*")
1488                          && !mer.startsWith("^")) {
1489                        konf_navn = konf_navn + " " + mer;
1490                        mer = inn.inWord();
1491                    }
1492                    konf[3] = konf_navn;
1493                    her = mer;
1494                }//end $a
1495                else if(her.startsWith("$c")) {
1496                    konf_sted = her.substring(2);
1497                    mer = inn.inWord();
1498                    while(!mer.startsWith("$") && !mer.startsWith("*")
1499                          && !mer.startsWith("^")) {
1500                        konf_sted = konf_sted + " " + mer;
1501                        mer = inn.inWord();
1502                    }
1503                    konf[4] = konf_sted;
1504                    her = mer;
1505                }//end $c
1506                else if(her.startsWith("$d")) {
1507                    konf_tid = her.substring(2);
1508                    mer = inn.inWord();
1509                    while(!mer.startsWith("$") && !mer.startsWith("*")
```

```
1510                    && !mer.startsWith("^")) {
1511                konf_tid = konf_tid + " " + mer;
1512                mer = inn.inWord();
1513            }
1514            konf[2] = konf_tid;
1515            her = mer;
1516        }//end $d
1517        else if(her.startsWith("$n")) {
1518            konf_nummer = her.substring(2);
1519            mer = inn.inWord();
1520            while(!mer.startsWith("$") && !mer.startsWith("*")
1521                  && !mer.startsWith("^")) {
1522                konf_nummer = konf_nummer + " " + mer;
1523                mer = inn.inWord();
1524            }
1525            konf[5] = konf_nummer;
1526            her = mer;
1527        }//end $n
1528        else {
1529            her = inn.inWord();
1530            while(!her.startsWith("$") && !her.startsWith("*")
1531                  && !her.startsWith("^")) {
1532                her = inn.inWord();
1533            }//end while
1534        }//end else
1535
1536    }//end while
1537
1538    ArrayList l = (ArrayList)bok.get("konfene");
1539    l.add(konf);
1540
1541    return her;
1542  }//end konferanse()
1543
1544
1545  public String korporasjon(int korp_kode) {
1546      //Lese inn navn, sted, dato og nr for korporasjon
1547
1548      String korp_her = null;
1549      String korp_mer = null;
1550      String korp_id = null;
1551      String korp_navn = null;
1552      String korp_sted = null;
1553      String korp_nr = null;
1554      String korp_tid = null;
1555      String korporasjon_kode = Integer.toString(korp_kode);
1556      String korp[] = new String[6];
1557
1558      korp[0] = korporasjon_kode;
1559
1560      korporasjonid++;
1561      korp[1] = Integer.toString(korporasjonid);
1562
1563      korp_her = inn.inWord();
```

```
1564            while(korp_her.startsWith("$") ||
1565                  korp_her.charAt(1) == '$') {
1566                if(korp_her.startsWith("$a")) {
1567                    korp_navn = korp_her.substring(2);
1568                    korp_mer = inn.inWord();
1569                    while(!korp_mer.startsWith("$")
1570                          && !korp_mer.startsWith("*")
1571                          && !korp_mer.startsWith("^")) {
1572                        korp_navn = korp_navn + " " + korp_mer;
1573                        korp_mer = inn.inWord();
1574                    }
1575                    korp[3] = korp_navn;
1576                    korp_her = korp_mer;
1577                }//end $a
1578                else if(korp_her.charAt(1) == '$') {
1579                    korp_navn = korp_her.substring(3);
1580                    korp_mer = inn.inWord();
1581                    while(!korp_mer.startsWith("$")
1582                          && !korp_mer.startsWith("*")
1583                          && !korp_mer.startsWith("^")) {
1584                        korp_navn = korp_navn + " " + korp_mer;
1585                        korp_mer = inn.inWord();
1586                    }
1587                    korp[3] = korp_navn;
1588                    korp_her = korp_mer;
1589                }//end 0$a
1590                else if(korp_her.startsWith("$c")) {
1591                    korp_sted = korp_her.substring(2);
1592                    korp_mer = inn.inWord();
1593                    while(!korp_mer.startsWith("$")
1594                          && !korp_mer.startsWith("*")
1595                          && !korp_mer.startsWith("^")) {
1596                        korp_sted = korp_sted + " " + korp_mer;
1597                        korp_mer = inn.inWord();
1598                    }
1599                    korp[4] = korp_sted;
1600                    korp_her = korp_mer;
1601                }//end $c
1602                else if(korp_her.startsWith("$d")) {
1603                    korp_tid = korp_her.substring(2);
1604                    korp_mer = inn.inWord();
1605                    while(!korp_mer.startsWith("$")
1606                          && !korp_mer.startsWith("*")
1607                          && !korp_mer.startsWith("^")) {
1608                        korp_tid = korp_tid + korp_mer;
1609                        korp_mer = inn.inWord();
1610                    }
1611                    korp[2] = korp_tid;
1612                    korp_her = korp_mer;
1613                }//end $d
1614                else if(korp_her.startsWith("$n")) {
1615                    korp_nr = korp_her.substring(2);
1616                    korp_mer = inn.inWord();
1617                    while(!korp_mer.startsWith("$")
```

```
1618                      && !korp_mer.startsWith("*")
1619                      && !korp_mer.startsWith("^")) {
1620                  korp_nr = korp_nr + korp_mer;
1621                  korp_mer = inn.inWord();
1622              }
1623              korp[5] = korp_nr;
1624              korp_her = korp_mer;
1625          }//end $n
1626          else {
1627              korp_her = inn.inWord();
1628              while(!korp_her.startsWith("$")
1629                    && !korp_her.startsWith("*")
1630                    && !korp_her.startsWith("^")) {
1631                  korp_her = inn.inWord();
1632              }//end while
1633          }//end else
1634      }//end while
1635
1636      ArrayList l = (ArrayList)bok.get("korpene");
1637      l.add(korp);
1638
1639      return korp_her;
1640  }//end korporasjon()
1641
1642
1643  public String tittel(int t_kode, String teg) {
1644      //Lese kode, hovedtittel, aarstall, utgivelsesaar, info
1645      //og legge inn i ArrayList titlene i HashMap bok
1646
1647      String hovedtittel = null;
1648      String tittel_aar = null;
1649      String tittel_utg_aar = null;
1650      String tittel_info = null;
1651      String a = null;
1652      String b = null;
1653      String tittel_kode = Integer.toString(t_kode);
1654      String tegnet = teg;
1655      String titt[] = new String[6];
1656
1657      titt[0] = tittel_kode;
1658
1659      tittelid++;
1660      titt[1] = Integer.toString(tittelid);
1661
1662      if(tittel_kode.equals("130") || tittel_kode.equals("240")
1663         || tittel_kode.equals("630") || tittel_kode.equals("730")
1664         || tittel_kode.equals("740") ||
1665         tittel_kode.equals("830")){
1666
1667          a = inn.inWord();
1668
1669          if(a.startsWith("0$")) {
1670              a = a.substring(1);
1671          }
```

```
1672
1673            while(a.startsWith("$")) {
1674
1675                if(a.startsWith("$a")) {
1676                    hovedtittel = a.substring(2);
1677                    b = inn.inWord();
1678                    while(!b.startsWith("$") && !b.startsWith("*")
1679                          && !b.startsWith("^")) {
1680                        hovedtittel = hovedtittel + " " + b;
1681                        b = inn.inWord();
1682                    }
1683                    titt[2] = hovedtittel;
1684                    a = b;
1685                }//end $a
1686                else if(a.startsWith("$b")) {
1687                    tittel_info = a.substring(2);
1688                    b = inn.inWord();
1689                    while(!b.startsWith("$") && !b.startsWith("*")
1690                          && !b.startsWith("^")) {
1691                        tittel_info = tittel_info + " " + b;
1692                        b = inn.inWord();
1693                    }
1694                    titt[5] = tittel_info;
1695                    a = b;
1696                }//end $b
1697                else if(a.startsWith("$d")) {
1698                    tittel_aar = a.substring(2);
1699                    b = inn.inWord();
1700                    while(!b.startsWith("$") && !b.startsWith("*")
1701                          && !b.startsWith("^")) {
1702                        tittel_aar = tittel_aar + " " + b;
1703                        b = inn.inWord();
1704                    }
1705                    titt[3] = tittel_aar;
1706                    a = b;
1707                }//end $d
1708                else if(a.startsWith("$f")) {
1709                    tittel_utg_aar = a.substring(2);
1710                    b = inn.inWord();
1711                    while(!b.startsWith("$") && !b.startsWith("*")
1712                          && !b.startsWith("^")) {
1713                        tittel_utg_aar = tittel_utg_aar + " " + b;
1714                        b = inn.inWord();
1715                    }
1716                    titt[4] = tittel_utg_aar;
1717                    a = b;
1718                }//end $f
1719                else {
1720                    a = inn.inWord();
1721                    while(!a.startsWith("$") && !a.startsWith("*")
1722                          && !a.startsWith("^")) {
1723                        a = inn.inWord();
1724                    }//end while
1725                }//end else
```

```
1726            }//end while()
1727        }//end if 130, 240, 630, 730, 740, 830
1728
1729        else if(tittel_kode.equals("245") ||
1730                tittel_kode.equals("246")){
1731
1732            //Hvis tegnet inneholder $a
1733            if(tegnet.indexOf("$a") >= 0) {
1734                a = tegnet.substring(tegnet.indexOf("$a"));
1735            }
1736            //Hvis *245 ikke har indikatorer
1737            else if(tegnet.equals("*245")) {
1738                a = inn.inWord();
1739            }
1740
1741            while(a.startsWith("$")) {
1742                if(a.startsWith("$a")) {
1743                    hovedtittel = a.substring(2);
1744                    b = inn.inWord();
1745                    while(!b.startsWith("$") && !b.startsWith("*")
1746                          && !b.startsWith("^")) {
1747                        hovedtittel = hovedtittel + " " + b;
1748                        b = inn.inWord();
1749                    }
1750                    titt[2] = hovedtittel;
1751                    a = b;
1752                }//end $a
1753                else if(a.startsWith("$b")) {
1754                    tittel_info = a.substring(2);
1755                    b = inn.inWord();
1756                    while(!b.startsWith("$") && !b.startsWith("*")
1757                          && !b.startsWith("^")) {
1758                        tittel_info = tittel_info + " " + b;
1759                        b = inn.inWord();
1760                    }
1761                    titt[5] = tittel_info;
1762                    a = b;
1763                }//end $b
1764                else if(a.startsWith("$d")) {
1765                    tittel_aar = a.substring(2);
1766                    b = inn.inWord();
1767                    while(!b.startsWith("$") && !b.startsWith("*")
1768                          && !b.startsWith("^")) {
1769                        tittel_aar = tittel_aar + " " + b;
1770                        b = inn.inWord();
1771                    }
1772                    titt[3] = tittel_aar;
1773                    a = b;
1774                }//end $d
1775                else if(a.startsWith("$f")) {
1776                    tittel_utg_aar = a.substring(2);
1777                    b = inn.inWord();
1778                    while(!b.startsWith("$") && !b.startsWith("*")
1779                          && !b.startsWith("^")) {
```

```
1780                        tittel_utg_aar = tittel_utg_aar + " " + b;
1781                        b = inn.inWord();
1782                    }
1783                    titt[4] = tittel_utg_aar;
1784                    a = b;
1785                }//end $f
1786                else {
1787                    a = inn.inWord();
1788                    while(!a.startsWith("$") && !a.startsWith("*")
1789                          && !a.startsWith("^")) {
1790                        a = inn.inWord();
1791                    }//end while
1792                }//end else
1793            }//end while()
1794        }//end else if 245, 246
1795
1796        ArrayList l = (ArrayList)bok.get("titlene");
1797        l.add(titt);
1798
1799        return a;
1800
1801    }//end tittel()
1802
1803
1804    public String forkortet_tittel() {
1805        //lese forkortet tittel og legge inn i bok
1806
1807        String een = null;
1808        String too = null;
1809        String forkortet_tittel = null;
1810
1811        een = inn.inWord();
1812        while(een.startsWith("$")) {
1813            if(een.startsWith("$a")) {
1814                forkortet_tittel = een.substring(2);
1815                too = inn.inWord();
1816                while(!too.startsWith("$") && !too.startsWith("*")
1817                      && !too.startsWith("^")) {
1818                    forkortet_tittel = forkortet_tittel + " " + too;
1819                    too = inn.inWord();
1820                }
1821                bok.put("forkortet_tittel",forkortet_tittel);
1822                een = too;
1823            }//end $a
1824            else {
1825                een = inn.inWord();
1826                while(!een.startsWith("$") && !een.startsWith("*")
1827                      && !een.startsWith("^")) {
1828                    een = inn.inWord();
1829                }//end while
1830            }//end else
1831        }//end while()
1832        return een;
1833    }//end forkortet_tittel()
```

```
1834
1835
1836        public String nokkel_tittel() {
1837            //lese forkortet tittel og tilfoyelse_fforklaring
1838
1839            String one = null;
1840            String two = null;
1841            String nokkel_tittel = null;
1842            String nforklaring = null;
1843
1844            one = inn.inWord();
1845            while(one.startsWith("$")) {
1846                if(one.startsWith("$a")) {
1847                    nokkel_tittel = one.substring(2);
1848                    two = inn.inWord();
1849                    while(!two.startsWith("$") && !two.startsWith("*")
1850                          && !two.startsWith("^")) {
1851                        nokkel_tittel = nokkel_tittel + " " + two;
1852                        two = inn.inWord();
1853                    }
1854
1855                    bok.put("nokkel_tittel",nokkel_tittel);
1856                    one = two;
1857                }//end $a
1858                else if(one.startsWith("$b")) {
1859                    nforklaring = one.substring(2);
1860                    two = inn.inWord();
1861                    while(!two.startsWith("$") && !two.startsWith("*")
1862                          && !two.startsWith("^")) {
1863                        nforklaring = nforklaring + " " + two;
1864                        two = inn.inWord();
1865                    }
1866                    bok.put("nforklaring",nforklaring);
1867                    one = two;
1868                }//end $b
1869                else {
1870                    one = inn.inWord();
1871                    while(!one.startsWith("$") && !one.startsWith("*")
1872                          && !one.startsWith("^")) {
1873                        one = inn.inWord();
1874                    }//end while
1875                }//end else
1876            }//end while()
1877            return one;
1878        }//end nokkel_tittel()
1879
1880        public String utgave() {
1881            //lese utgave og ansvarshavende og legge inn i bok.
1882
1883            String e = null;
1884            String f = null;
1885            String utgaven_av = null;
1886            String navnet_ansvarlig = null;
1887
```

```
1888            e = inn.inWord();
1889            while(e.startsWith("$")) {
1890                if(e.startsWith("$a")) {
1891                    utgaven_av = e.substring(2);
1892                    f = inn.inWord();
1893                    while(!f.startsWith("$") && !f.startsWith("*")
1894                          && !f.startsWith("^")) {
1895                        utgaven_av = utgaven_av + " " + f;
1896                        f = inn.inWord();
1897                    }
1898                    bok.put("utgaven_av",utgaven_av);
1899                    e = f;
1900                }//end $a
1901                else if(e.startsWith("$b")) {
1902                    navnet_ansvarlig = e.substring(2);
1903                    f = inn.inWord();
1904                    while(!f.startsWith("$") && !f.startsWith("*")
1905                          && !f.startsWith("^")) {
1906                        navnet_ansvarlig = navnet_ansvarlig + " " + f;
1907                        f = inn.inWord();
1908                    }
1909                    bok.put("navnet_ansvarlig",navnet_ansvarlig);
1910                    e = f;
1911                }//end $b
1912                else {
1913                    e = inn.inWord();
1914                    while(!e.startsWith("$") && !e.startsWith("*")
1915                          && !e.startsWith("^")) {
1916                        e = inn.inWord();
1917                    }//end while
1918                }//end else
1919            }//end while()
1920            return e;
1921        }//end utgave()
1922
1923
1924        public String utgivelse() {
1925            //Lese navn, år, trykkerinavn, trykkeristed og utgivelsessted
1926            //for utgivelse fra fila og legge det inn i bok
1927
1928            String utg_sted = null;
1929            String navnet_forlag = null;
1930            String utg_aar = null;
1931            String navnet_trykkested = null;
1932            String navnet_trykkeri = null;
1933            String en = null;
1934            String to = null;
1935
1936            en = inn.inWord();
1937            while(en.startsWith("$")) {
1938                if(en.startsWith("$a")) {
1939                    utg_sted = en.substring(2);
1940                    to = inn.inWord();
1941                    while(!to.startsWith("$") && !to.startsWith("*")
```

```
1942                    && !to.startsWith("^")) {
1943                utg_sted = utg_sted + " " + to;
1944                to = inn.inWord();
1945            }
1946            bok.put("utg_sted",utg_sted);
1947            en = to;
1948        }//end $a
1949        else if(en.startsWith("$b")) {
1950            navnet_forlag = en.substring(2);
1951            to = inn.inWord();
1952            while(!to.startsWith("$") && !to.startsWith("*")
1953                    && !to.startsWith("^")) {
1954                navnet_forlag = navnet_forlag + " " + to;
1955                to = inn.inWord();
1956            }
1957            bok.put("navnet_forlag",navnet_forlag);
1958            en = to;
1959        }//end $b
1960        else if(en.startsWith("$c")) {
1961            utg_aar = en.substring(2);
1962            to = inn.inWord();
1963            while(!to.startsWith("$") && !to.startsWith("*")
1964                    && !to.startsWith("^")) {
1965                utg_aar = utg_aar + " " + to;
1966                to = inn.inWord();
1967            }
1968            bok.put("utg_aar",utg_aar);
1969            en = to;
1970        }//end $c
1971        else if(en.startsWith("$e")) {
1972            navnet_trykkested = en.substring(2);
1973            to = inn.inWord();
1974            while(!to.startsWith("$") && !to.startsWith("*")
1975                    && !to.startsWith("^")) {
1976                navnet_trykkested = navnet_trykkested + " " + to;
1977                to = inn.inWord();
1978            }
1979            bok.put("navnet_trykkested",navnet_trykkested);
1980            en = to;
1981        }//end $e
1982        else if(en.startsWith("$f")) {
1983            navnet_trykkeri = en.substring(2);
1984            to = inn.inWord();
1985            while(!to.startsWith("$") && !to.startsWith("*")
1986                    && !to.startsWith("^")) {
1987                navnet_trykkeri = navnet_trykkeri + " " + to;
1988                to = inn.inWord();
1989            }
1990            bok.put("navnet_trykkeri",navnet_trykkeri);
1991            en = to;
1992        }//end $f
1993        else {
1994            en = inn.inWord();
1995            while(!en.startsWith("$") && !en.startsWith("*")
```

```
1996                    && !to.startsWith("^")) {
1997                en = inn.inWord();
1998            }//end while
1999        }//end else
2000    }//end while
2001    return en;
2002 }//end utgivelse()
2003
2004
2005 public String omfang() {
2006    //lese omfang og legge inn i bok.
2007
2008    String aa = null;
2009    String bb = null;
2010    String omfang = null;
2011
2012    aa = inn.inWord();
2013    while(aa.startsWith("$")) {
2014        if(aa.startsWith("$a")) {
2015            omfang = aa.substring(2);
2016            bb = inn.inWord();
2017            while(!bb.startsWith("$") && !bb.startsWith("*")
2018                  && !bb.startsWith("^")) {
2019                omfang = omfang + " " + bb;
2020                bb = inn.inWord();
2021            }
2022            bok.put("omfang", omfang);
2023            aa = bb;
2024        }//end $a
2025        else {
2026            aa = inn.inWord();
2027            while(!aa.startsWith("$") && !aa.startsWith("*")
2028                  && !aa.startsWith("^")) {
2029                aa = inn.inWord();
2030            }//end while
2031        }//end else
2032    }//end while()
2033    return aa;
2034 }//end omfang()
2035
2036
2037 public String periodisitet() {
2038    //lese periodisitet og legge inn i bok.
2039
2040    String aaa = null;
2041    String bbb = null;
2042    String frekvens_utgivelse = null;
2043
2044    aaa = inn.inWord();
2045    while(aaa.startsWith("$")) {
2046        if(aaa.startsWith("$a")) {
2047            frekvens_utgivelse = aaa.substring(2);
2048            bbb = inn.inWord();
2049            while(!bbb.startsWith("$") && !bbb.startsWith("*")
```

```
2050                      && !bbb.startsWith("^")) {
2051                  frekvens_utgivelse =
2052                      frekvens_utgivelse + " " + bbb;
2053                  bbb = inn.inWord();
2054              }
2055              bok.put("frekvens_utgivelse", frekvens_utgivelse);
2056              aaa = bbb;
2057          }//end $a
2058          else {
2059              aaa = inn.inWord();
2060              while(!aaa.startsWith("$") && !aaa.startsWith("*")
2061                    && !aaa.startsWith("^")) {
2062                  aaa = inn.inWord();
2063              }//end while
2064          }//end else
2065      }//end while()
2066      return aaa;
2067  }//end periodisitet()
2068
2069
2070  public String note() {
2071      //Lese generell note og legge inn i bok.
2072
2073      String ee = null;
2074      String ff = null;
2075      String noten_for = null;
2076
2077      ee = inn.inWord();
2078      while(ee.startsWith("$")) {
2079          if(ee.startsWith("$a")) {
2080              noten_for = ee.substring(2);
2081              ff = inn.inWord();
2082              while(!ff.startsWith("$") && !ff.startsWith("*")
2083                    && !ff.startsWith("^")) {
2084                  noten_for = noten_for + " " + ff;
2085                  ff = inn.inWord();
2086              }
2087
2088              ArrayList l = (ArrayList)bok.get("noter_for");
2089              l.add(noten_for);
2090
2091              ee = ff;
2092          }//end $a
2093          else {
2094              ee = inn.inWord();
2095              while(!ee.startsWith("$") && !ee.startsWith("*")
2096                    && !ee.startsWith("^")) {
2097                  ee = inn.inWord();
2098              }//end while
2099          }//end else
2100      }//end while()
2101      return ee;
2102  }//end note()
2103
```

```
2104
2105    public String dokeier() {
2106        //Lese inn navn, ant, filial og hylle for dokeier
2107
2108        String dok_her = null;
2109        String dok_mer = null;
2110        String eier_bibl = null;
2111        String filial = null;
2112        String hylle = null;
2113        String ant_eks = null;
2114
2115        dok_her = inn.inWord();
2116
2117        while(dok_her.startsWith("$")) {
2118            if(dok_her.startsWith("$n")) {
2119                //$a er da midt inne i dok_her
2120                //Må finne $a og lese inn eier_bibl
2121                if(dok_her.indexOf("$a") >= 0) {
2122                    eier_bibl = dok_her.
2123                        substring(dok_her.indexOf("$a"));
2124                    eier_bibl = eier_bibl.substring(2);
2125                }
2126                dok_mer = inn.inWord();
2127                while(!dok_mer.startsWith("$")
2128                      && !dok_mer.startsWith("*")
2129                      && !dok_mer.startsWith("^")) {
2130                    eier_bibl = eier_bibl + " " + dok_mer;
2131                    dok_mer = inn.inWord();
2132                }
2133                bok.put("eier_bibl",eier_bibl);
2134                dok_her = dok_mer;
2135            }//end $n
2136            else if(dok_her.startsWith("$b")) {
2137                filial = dok_her.substring(2);
2138                dok_mer = inn.inWord();
2139                while(!dok_mer.startsWith("$")
2140                      && !dok_mer.startsWith("*")
2141                      && !dok_mer.startsWith("^")) {
2142                    filial = filial + " " + dok_mer;
2143                    dok_mer = inn.inWord();
2144                }
2145                filial = filial.trim();
2146                bok.put("filial",filial);
2147                dok_her = dok_mer;
2148            }//end $b
2149            else if(dok_her.startsWith("$c")) {
2150                hylle = dok_her.substring(2);
2151                dok_mer = inn.inWord();
2152                while(!dok_mer.startsWith("$")
2153                      && !dok_mer.startsWith("*")
2154                      && !dok_mer.startsWith("^")) {
2155                    hylle = hylle + " " + dok_mer;
2156                    dok_mer = inn.inWord();
2157                }
```

```
2158                hylle = hylle.trim();
2159                bok.put("hylle",hylle);
2160                dok_her = dok_mer;
2161            }//end $c
2162            else if(dok_her.startsWith("$e")) {
2163                ant_eks = dok_her.substring(2);
2164                dok_mer = inn.inWord();
2165                while(!dok_mer.startsWith("$")
2166                      && !dok_mer.startsWith("*")
2167                      && !dok_mer.startsWith("^")) {
2168                    ant_eks = ant_eks + " " + dok_mer;
2169                    dok_mer = inn.inWord();
2170                }
2171                ant_eks = ant_eks.trim();
2172                bok.put("ant_eks",ant_eks);
2173                dok_her = dok_mer;
2174            }//end $n
2175            else {
2176                dok_her = inn.inWord();
2177                while(!dok_her.startsWith("$")
2178                      && !dok_her.startsWith("*")
2179                      && !dok_her.startsWith("^")) {
2180                    dok_her = inn.inWord();
2181                }//end while
2182            }//end else
2183        }//end while
2184        return dok_her;
2185    }//end dokeier
2186
2187
2188    public String emneord() {
2189        //Lese emneord
2190        String emneord = null;
2191        String ord = null;
2192        String neste = null;
2193
2194        ord = inn.inWord();
2195        while(ord.startsWith("$")) {
2196            if(ord.startsWith("$a")) {
2197                emneord = ord.substring(2);
2198                neste = inn.inWord();
2199                while(!neste.startsWith("$")
2200                      && !neste.startsWith("*")
2201                      && !neste.startsWith("^")) {
2202                    emneord = emneord + " " + neste;
2203                    neste = inn.inWord();
2204                }
2205                ArrayList l = (ArrayList)bok.get("emneordene");
2206                if(!l.contains(emneord)) {
2207                    l.add(emneord);
2208                }//end if
2209                ord = neste;
2210            }//end $a
2211            else {
```

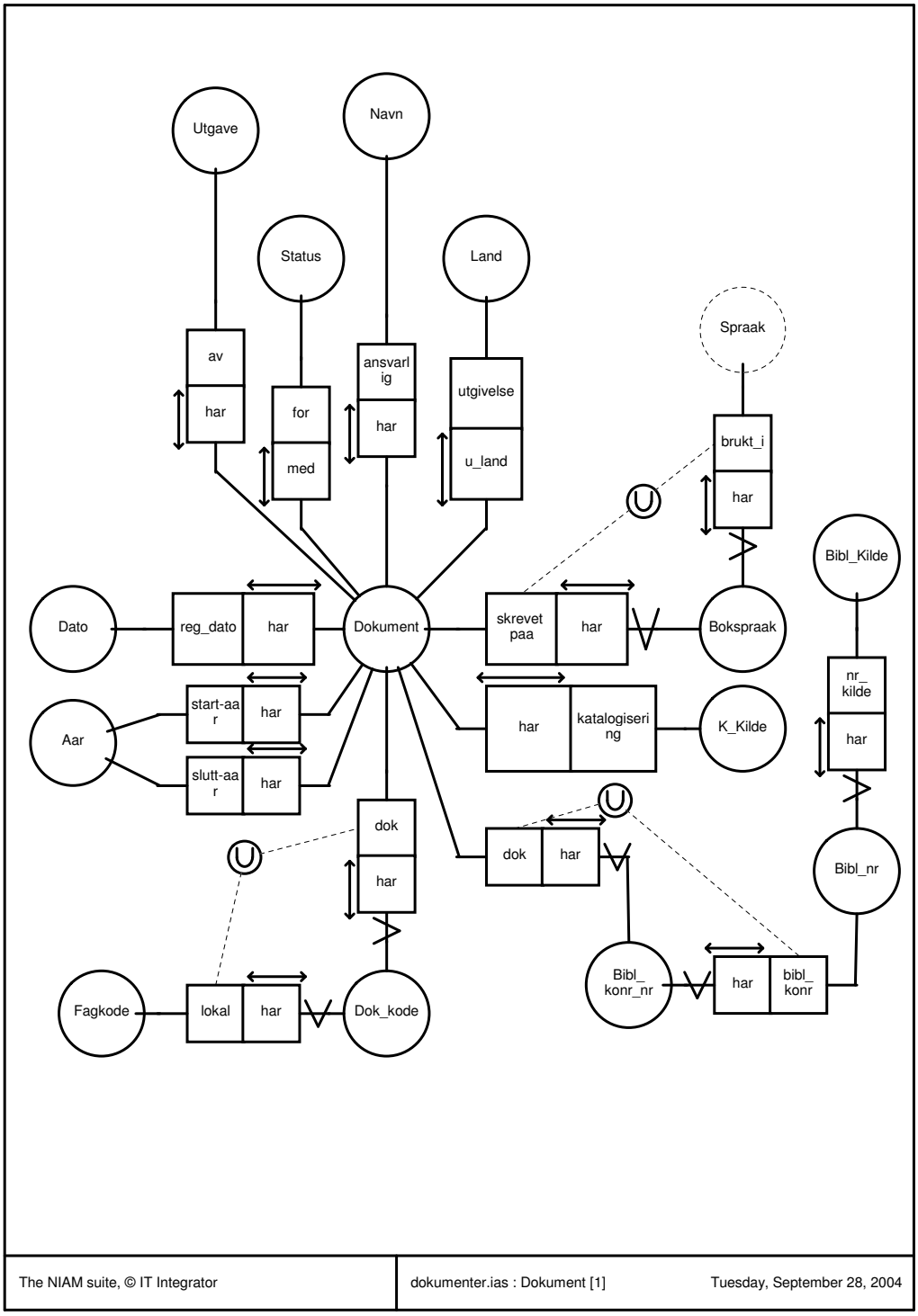
```
2212                ord = inn.inWord();
2213                while(!ord.startsWith("$") && !ord.startsWith("*")
2214                      && !ord.startsWith("^")) {
2215                    ord = inn.inWord();
2216                }//end while
2217            }//end else
2218        }//end while
2219        return ord;
2220    }//end emneord
2221
2222
2223    public String nokkelord() {
2224        //Lese nokkelord
2225        String nokkelord = null;
2226        String ordet = null;
2227        String nestemann = null;
2228
2229        ordet = inn.inWord();
2230        while(ordet.startsWith("$")) {
2231            if(ordet.startsWith("$a")) {
2232                nokkelord = ordet.substring(2);
2233                ArrayList l = (ArrayList)bok.get("nokkelordene");
2234                if(!l.contains(nokkelord)) {
2235                    l.add(nokkelord);
2236                }//end if
2237
2238                nestemann = inn.inWord();
2239                while(!nestemann.startsWith("$")
2240                      && !nestemann.startsWith("*")
2241                      && !nestemann.startsWith("^")) {
2242                    nokkelord = nestemann;
2243
2244                    if(!l.contains(nokkelord)) {
2245                        l.add(nokkelord);
2246                    }//end if
2247                    nestemann = inn.inWord();
2248                }
2249                ordet = nestemann;
2250            }//end $a
2251            else {
2252                ordet = inn.inWord();
2253                while(!ordet.startsWith("$")
2254                      && !ordet.startsWith("*")
2255                      && !ordet.startsWith("^")) {
2256                    ordet = inn.inWord();
2257                }//end while
2258            }//end else
2259        }//end while
2260        return ordet;
2261    }//end nokkelord()
2262
2263
2264    public String lokale_ord() {
2265        //Lese lokale emneord
```

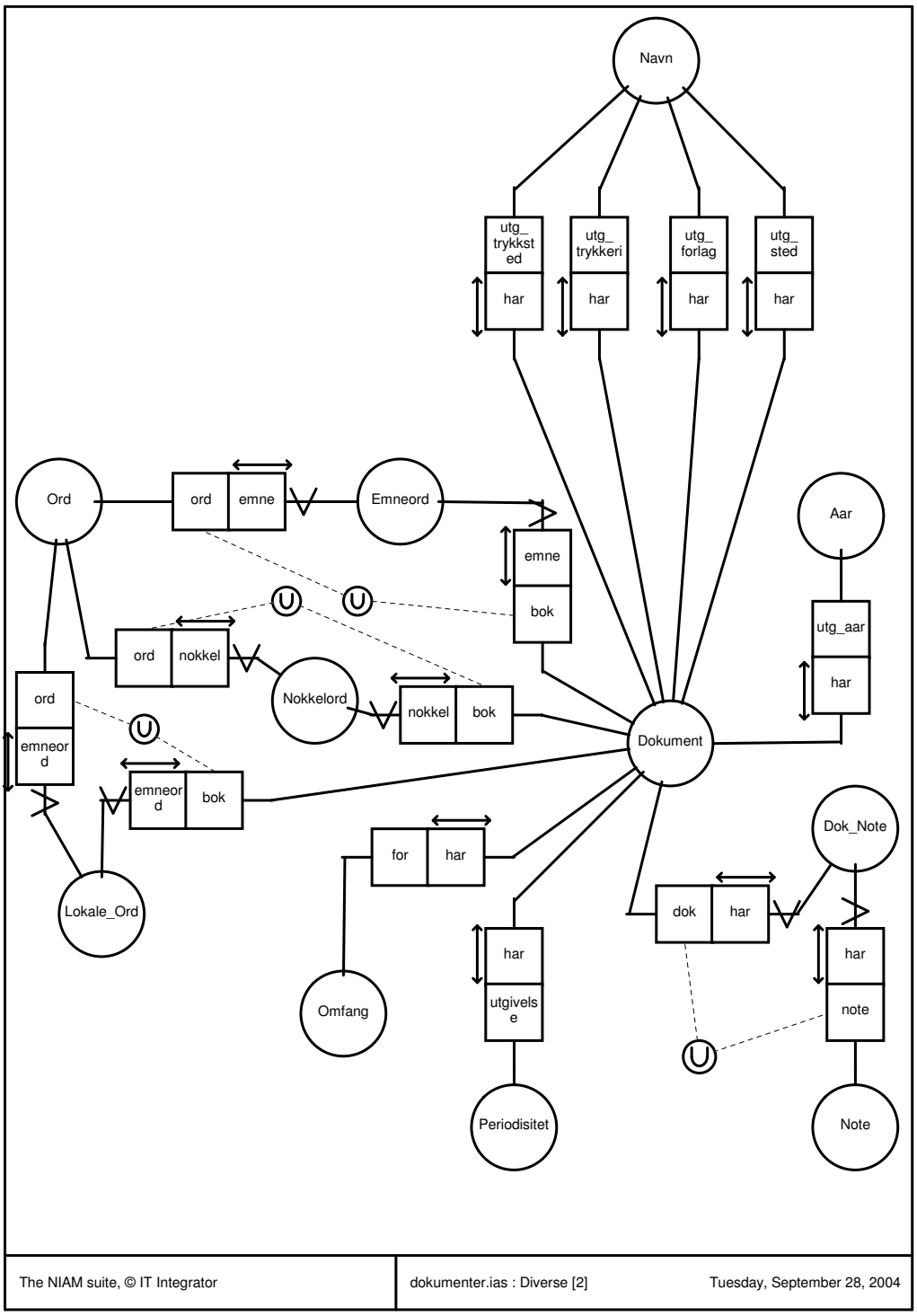
```
2266            String lokale_ord = null;
2267            String litt = null;
2268            String litt_til = null;
2269
2270            litt = inn.inWord();
2271            while(litt.startsWith("$")) {
2272                if(litt.startsWith("$a")) {
2273                    lokale_ord = litt.substring(2);
2274                    litt_til = inn.inWord();
2275                    while(!litt_til.startsWith("$")
2276                          && !litt_til.startsWith("*")
2277                          && !litt_til.startsWith("^")) {
2278                        lokale_ord = lokale_ord + " " + litt_til;
2279                        litt_til = inn.inWord();
2280                    }
2281
2282                    ArrayList l = (ArrayList)bok.get("lokale_ordene");
2283                    //Må sjekke at ikke samme ord forekommer flere ganger
2284                    if(!l.contains(lokale_ord)){
2285                        l.add(lokale_ord);
2286                    }//end if
2287
2288                    litt = litt_til;
2289                }//end $a
2290                else {
2291                    litt = inn.inWord();
2292                    while(!litt.startsWith("$") && !litt.startsWith("*")
2293                          && !litt.startsWith("^")) {
2294                        litt = inn.inWord();
2295                    }//end while
2296                }//end else
2297            }//end while
2298            return litt;
2299        }//end lokale_ord()
2300
2301
2302
2303        public String url() {
2304            //Lese vertsmaskin, elektronisk format, filstørrelse, url,
2305            //note og intern note fra fila og legge det i bok
2306
2307            String format = null;
2308            String filstorrelse = null;
2309            String url = null;
2310            String note = null;
2311            String intern_note = null;
2312            String liten = null;
2313            String stor = null;
2314
2315            liten = inn.inWord();
2316            while(liten.startsWith("$")) {
2317                if(liten.startsWith("$q")) {
2318                    format = liten.substring(2);
2319                    stor = inn.inWord();
```

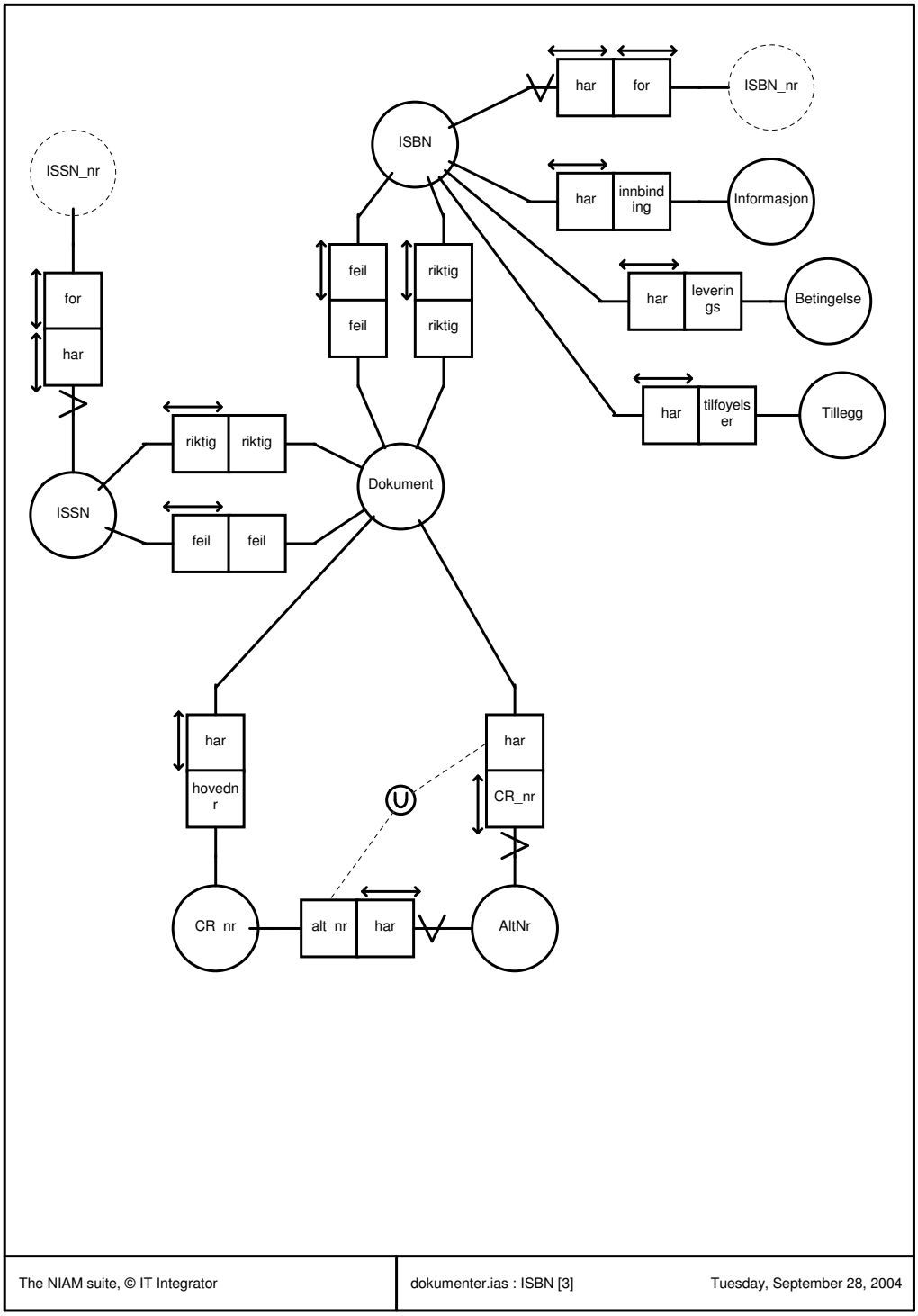
```
2320                while(!stor.startsWith("$") && !stor.startsWith("*")
2321                      && !stor.startsWith("^")) {
2322                    format = format + " " + stor;
2323                    stor = inn.inWord();
2324                }
2325                bok.put("format", format);
2326                liten = stor;
2327            }//end $q
2328            else if(liten.startsWith("$s")) {
2329                filstorrelse = liten.substring(2);
2330                stor = inn.inWord();
2331                while(!stor.startsWith("$") && !stor.startsWith("*")
2332                      && !stor.startsWith("^")) {
2333                    filstorrelse = filstorrelse + " " + stor;
2334                    stor = inn.inWord();
2335                }
2336                bok.put("filstorrelse", filstorrelse);
2337                liten = stor;
2338            }//end $s
2339            else if(liten.startsWith("$u")) {
2340                url = liten.substring(2);
2341                stor = inn.inWord();
2342                while(!stor.startsWith("$") && !stor.startsWith("*")
2343                      && !stor.startsWith("^")) {
2344                    url = url + stor;
2345                    stor = inn.inWord();
2346                }
2347                ArrayList l = (ArrayList)bok.get("urlene");
2348                l.add(url);
2349
2350                liten = stor;
2351            }//end $u
2352            else if(liten.startsWith("$z")) {
2353                note = liten.substring(2);
2354                stor = inn.inWord();
2355                while(!stor.startsWith("$") && !stor.startsWith("*")
2356                      && !stor.startsWith("^")) {
2357                    note = note + " " + stor;
2358                    stor = inn.inWord();
2359                }
2360                bok.put("note",note);
2361                liten = stor;
2362            }//end $z
2363            else if(liten.startsWith("$x")) {
2364                intern_note = liten.substring(2);
2365                stor = inn.inWord();
2366                while(!stor.startsWith("$") && !stor.startsWith("*")
2367                      && !stor.startsWith("^")) {
2368                    intern_note = intern_note + " " + stor;
2369                    stor = inn.inWord();
2370                }
2371                bok.put("intern_note",intern_note);
2372                liten = stor;
2373            }//end $x
```

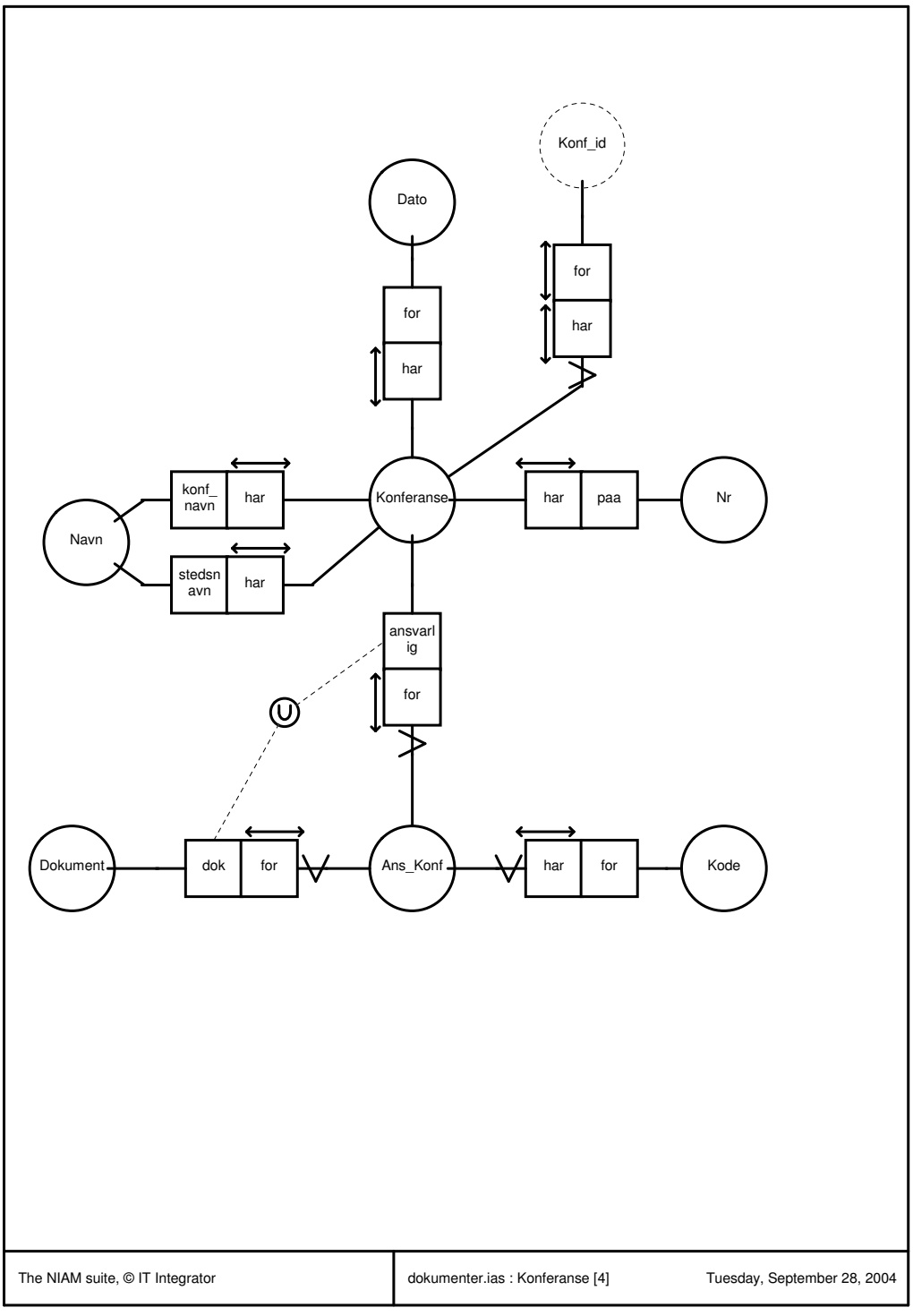
```
2374                else {
2375                    liten = inn.inWord();
2376                    while(!liten.startsWith("$")
2377                          && !liten.startsWith("*")
2378                          && !stor.startsWith("^")) {
2379                        liten = inn.inWord();
2380                    }//end while
2381                }//end else
2382            }//end while
2383            return liten;
2384        }//end url()
2385
2386
2387        public static void main( String argv[] ) throws Exception {
2388
2389            if(argv.length == 1){
2390                new lese(argv[0]);
2391            }
2392            else {
2393                System.out.println("Bruk: \n \t java lese 'filnavn'");
2394            }
2395
2396        }//end main
2397
2398    }//end class lese
```

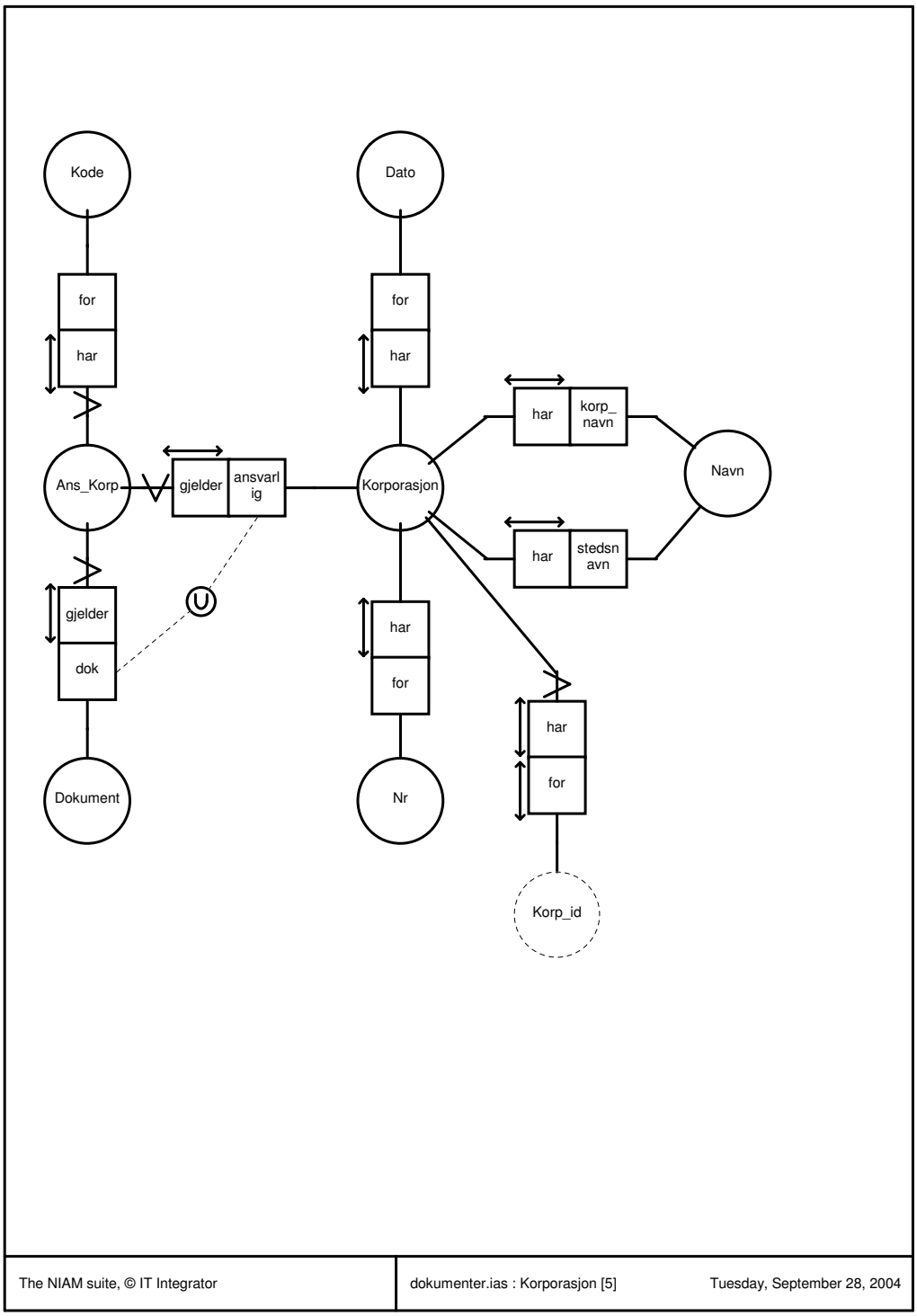
B NIAM-modell for den første relasjonelle databasen

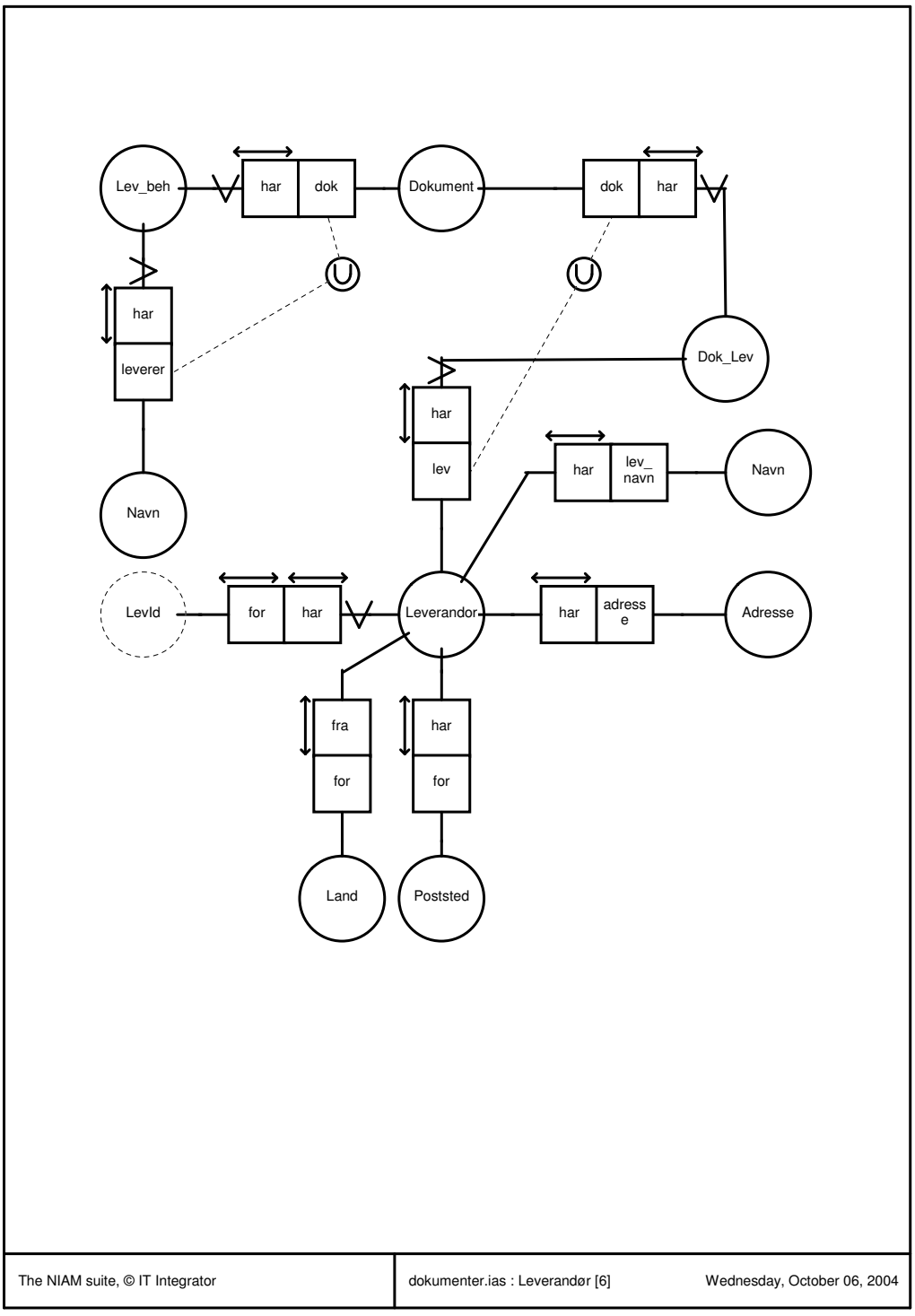


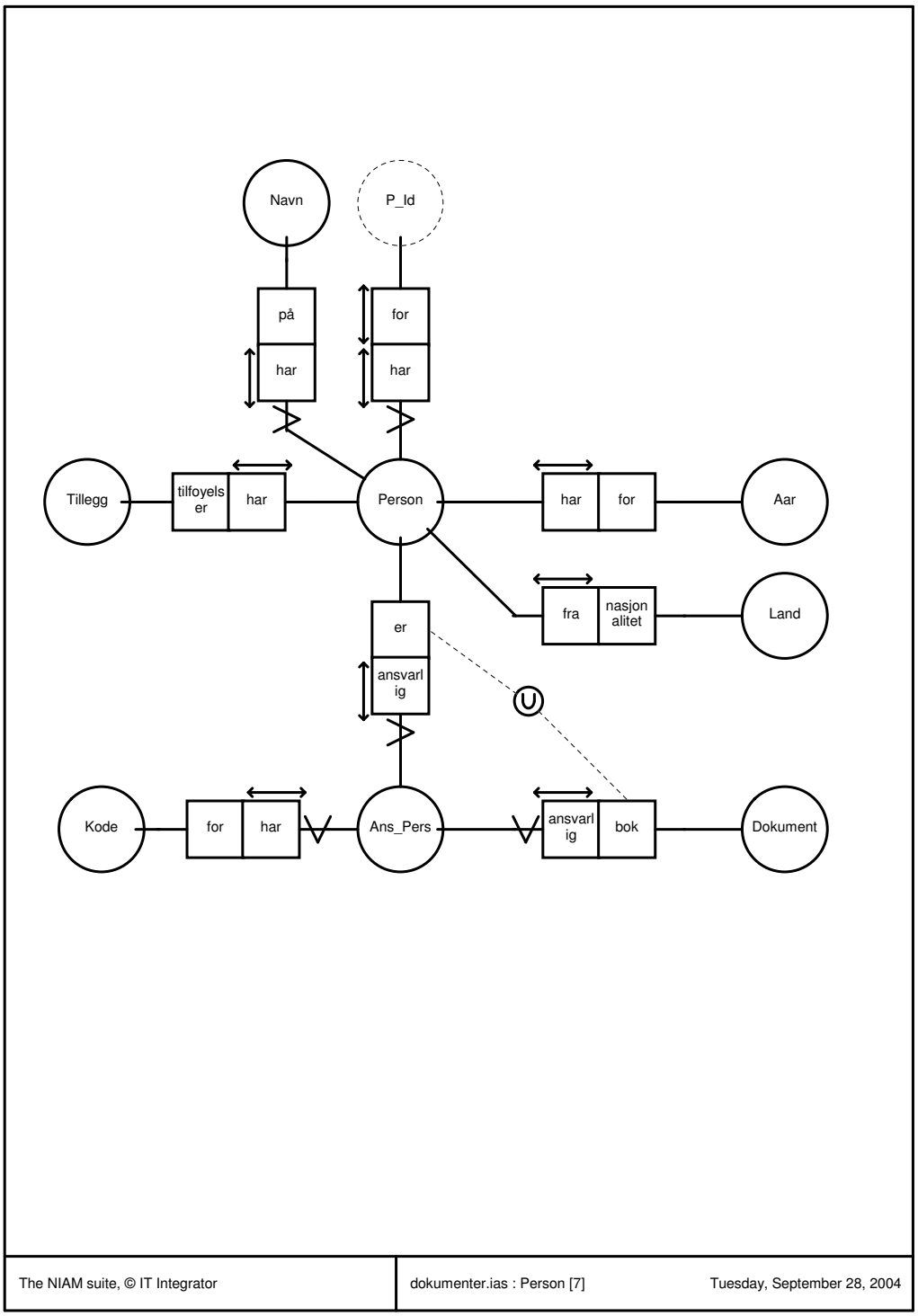

Navn
utg_ trykkst ed
utg_ trykkeri
utg_ forlag
utg_ sted
har
har
har
har
Ord
ord
emne
Emneord
emne
bok
Aar
utg_aar
har
ord
nokkel
Nokkelord
nokkel
bok
ord
emneor d
emneor d
bok
Dokument
Dok_Note
for
har
dok
har
har
note
Lokale_Ord
Omfang
har
utgivels e
Periodisitet
Note
The NIAM suite, © IT Integrator
dokumenter.ias : Diverse [2]
Tuesday, September 28, 2004

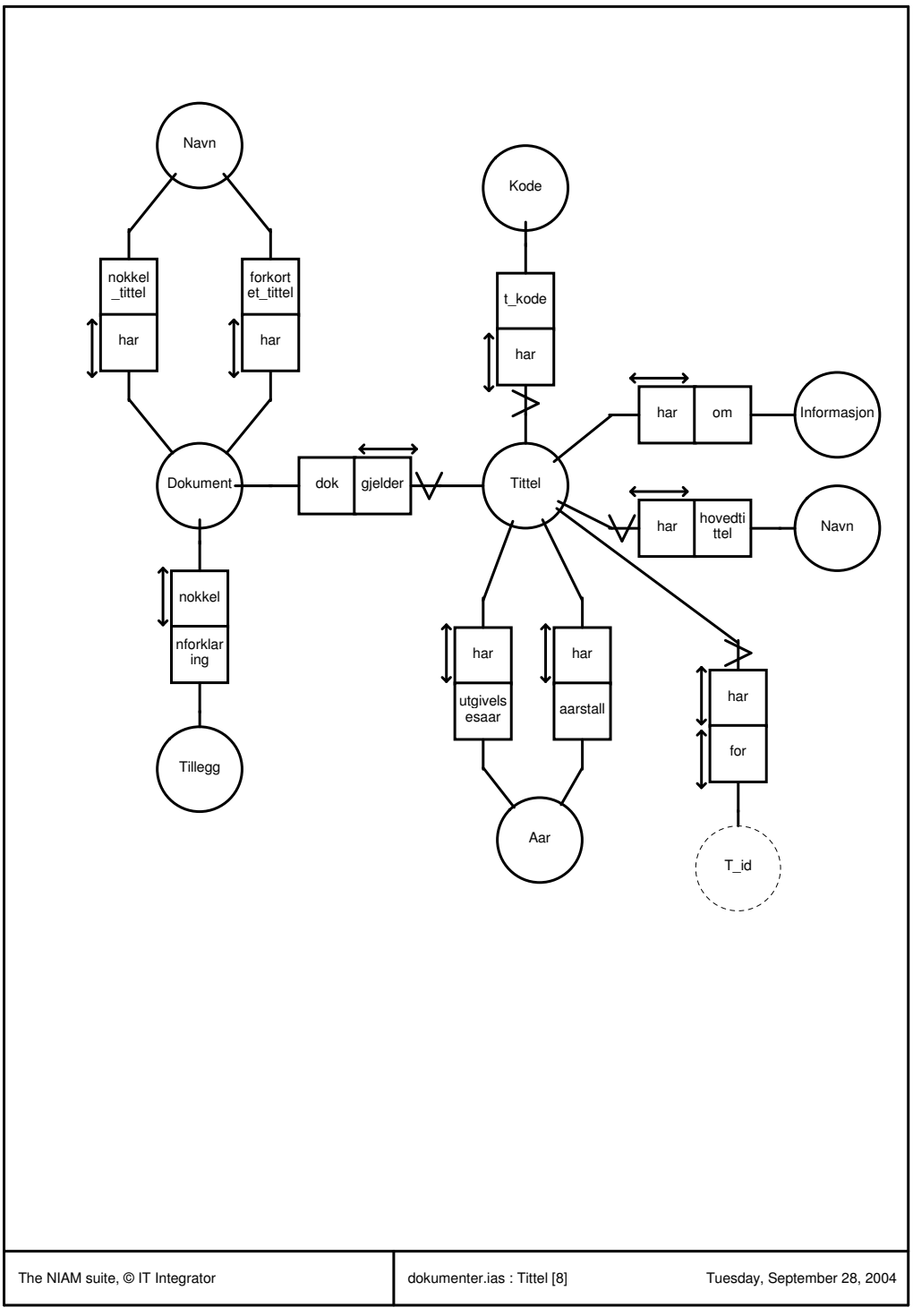

ISBN
har
for
ISBN_nr
har
innbind
ing
Informasjon
har
leverin
gs
Betingelse
har
tilfoyels
er
Tillegg
feil
feil
riktig
riktig
ISSN_nr
for
har
riktig
riktig
ISSN
feil
feil
Dokument
har
hovedn
r
har
CR_nr
U
CR_nr
alt_nr
har
AltNr
The NIAM suite, © IT Integrator
dokumenter.ias : ISBN [3]
Tuesday, September 28, 2004

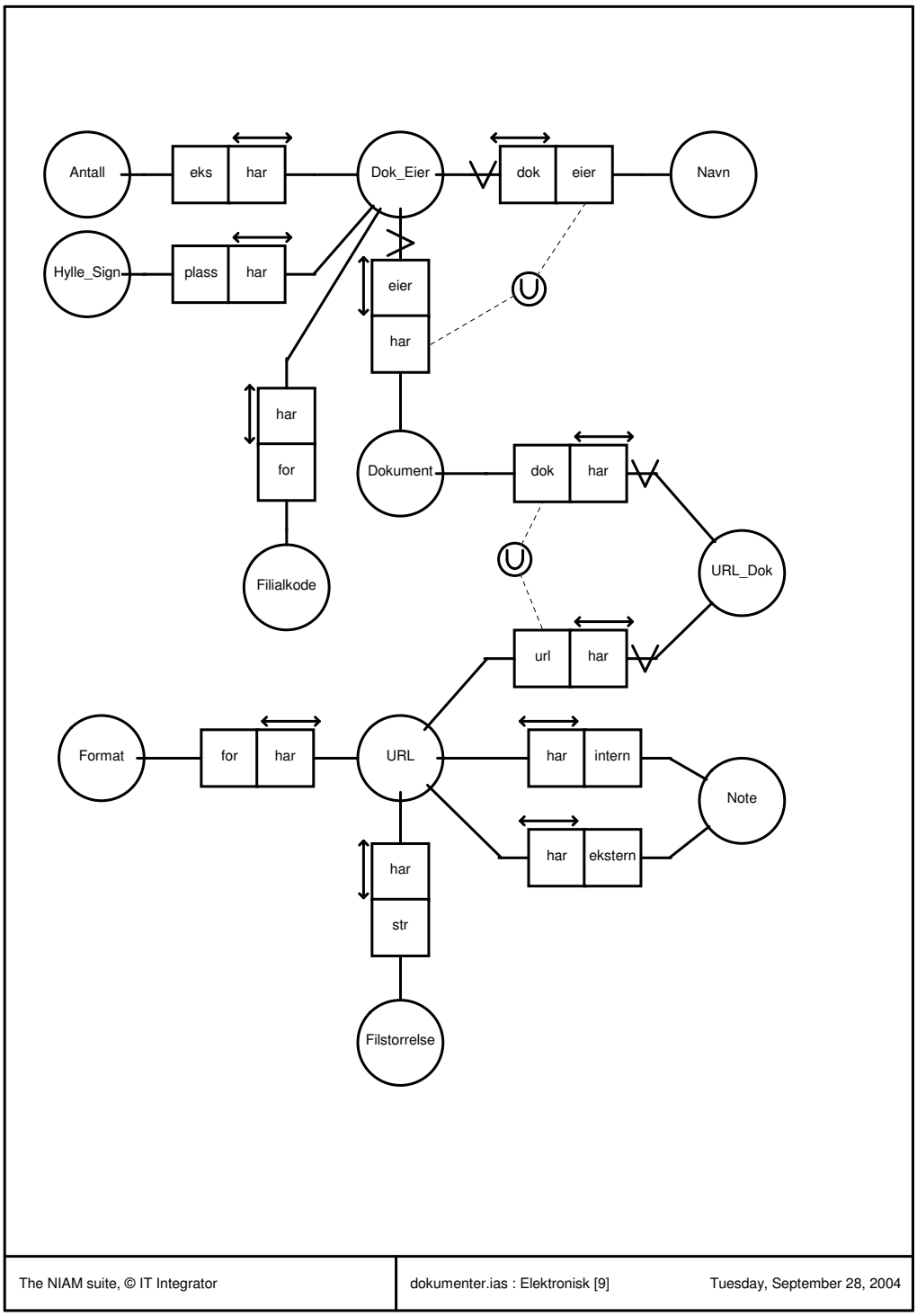

Konf_id
Dato
for
har
for
har
konf_
navn
har
Konferanse
har
paa
Nr
Navn
stedsn
avn
har
ansvarl
ig
for
U
Dokument
dok
for
Ans_Konf
har
for
Kode
The NIAM suite, © IT Integrator
dokumenter.ias : Konferanse [4]
Tuesday, September 28, 2004

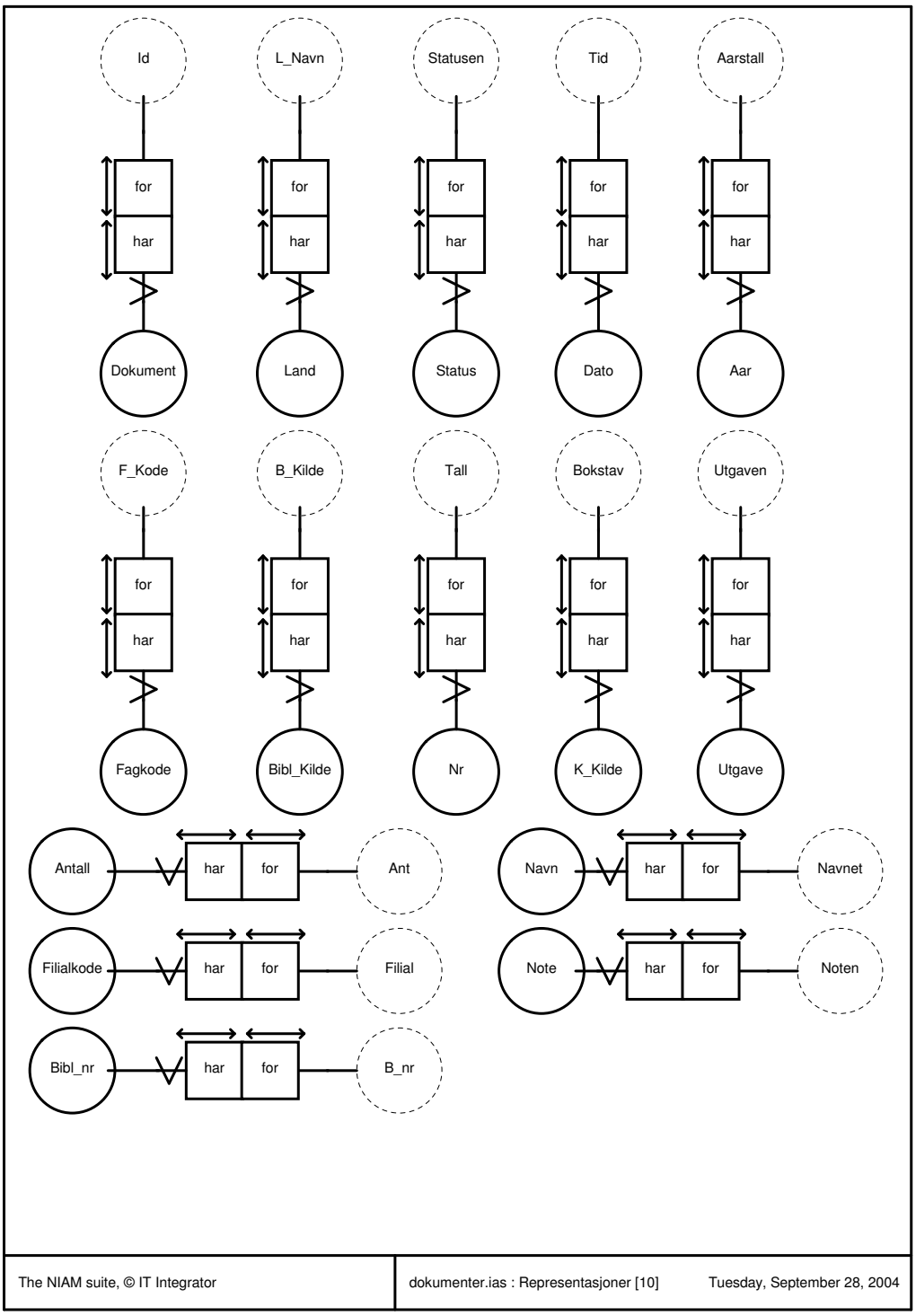

Kode
Dato
for
har
for
har
har
korp_
navn
Navn
Ans_Korp
gjelder
ansvarl
ig
Korporasjon
har
stedsn
avn
gjelder
dok
har
for
har
for
Dokument
Nr
Korp_id
The NIAM suite, © IT Integrator
dokumenter.ias : Korporasjon [5]
Tuesday, September 28, 2004

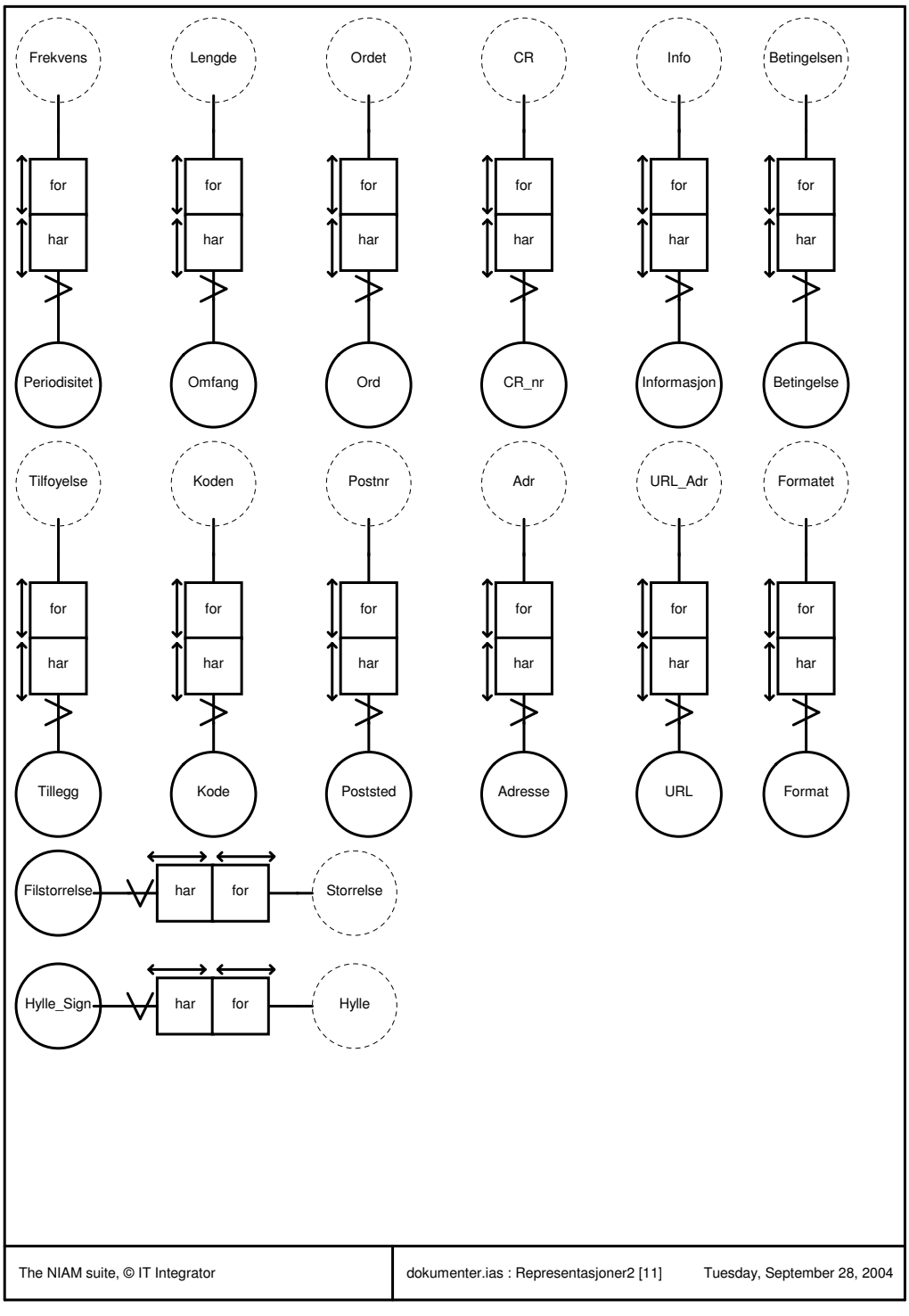

Lev_beh
har
dok
Dokument
dok
har
har
leverer
Dok_Lev
har
lev
har
lev_
navn
Navn
Navn
LevId
for
har
Leverandør
har
adress
e
Adresse
fra
for
har
for
Land
Poststed
The NIAM suite, © IT Integrator
dokumenter.ias : Leverandør [6]
Wednesday, October 06, 2004


Navn
P_Id
på
har
for
har
Tillegg
tilfoyelser
har
Person
har
for
Aar
fra
nasjonalitet
Land
er
ansvarlig
Kode
for
har
Ans_Pers
ansvarlig
bok
Dokument
The NIAM suite, © IT Integrator
dokumenter.ias : Person [7]
Tuesday, September 28, 2004


Navn
nokkel _tittel
har
forkort et_tittel
har
Dokument
dok
gjelder
Kode
t_kode
har
Tittel
har
om
Informasjon
har
hovedti ttel
Navn
nokkel
nforklar ing
Tillegg
har
utgivels esaar
har
aarstall
Aar
har
for
T_id
The NIAM suite, © IT Integrator
dokumenter.ias : Tittel [8]
Tuesday, September 28, 2004


Antall
eks
har
Dok_Eier
dok
eier
Navn
Hylle_Sign
plass
har
eier
har
har
for
Dokument
dok
har
Filialkode
URL_Dok
url
har
Format
for
har
URL
har
intern
Note
har
ekstern
har
str
Filstorrelse
The NIAM suite, © IT Integrator
dokumenter.ias : Elektronisk [9]
Tuesday, September 28, 2004


Id
L_Navn
Statusen
Tid
Aarstall
for
har
Dokument
Land
Status
Dato
Aar
F_Kode
B_Kilde
Tall
Bokstav
Utgaven
Fagkode
Bibl_Kilde
Nr
K_Kilde
Utgave
Antall
Ant
Navn
Navnet
Filialkode
Filial
Note
Noten
Bibl_nr
B_nr
The NIAM suite, © IT Integrator
dokumenter.ias : Representasjoner [10]
Tuesday, September 28, 2004


Frekvens
Lengde
Ordet
CR
Info
Betingelsen
for
har
Periodisitet
Omfang
Ord
CR_nr
Informasjon
Betingelse
Tilfoyelse
Koden
Postnr
Adr
URL_Adr
Formatet
Tillegg
Kode
Poststed
Adresse
URL
Format
Filstorrelse
Storrelse
Hylle_Sign
Hylle
The NIAM suite, © IT Integrator
dokumenter.ias : Representasjoner2 [11]
Tuesday, September 28, 2004

C Oracleskjema for den første relasjonelle databasen

```
Rem ****************************************************************
Rem
Rem Oracle 7 Database Description File for Gruppering
Rem Generation Date is 2004/5/28
Rem Model Number is 19
Rem
Rem (The NIAM Suite 3.4, (c)COPYRIGHT IT Integrator A/S. ALL RIGHTS RESERVED)
Rem
Rem (Gene date 08.09.00)
Rem ****************************************************************

Rem ***
Rem ***  Table Division for the Data Model
Rem ***

Create Table AltNr (
  Id_har                              NUMBER(18) NOT NULL,
  CR_alt_nr                           VARCHAR2 (100) NOT NULL)
 Storage (INITIAL 7K NEXT 7K
     PCTINCREASE 1)
/

Create Table Ans_Konf (
  Id_dok                              NUMBER(18) NOT NULL,
  Konf_id_ansvarlig                   VARCHAR2 (20) NOT NULL,
  Koden_for                           VARCHAR2 (10) NOT NULL)
 Storage (INITIAL 5K NEXT 5K
     PCTINCREASE 1)
/

Create Table Ans_Korp (
  Id_dok                              NUMBER(18) NOT NULL,
  Korp_id_ansvarlig                   VARCHAR2 (20) NOT NULL,
  Koden_for                           VARCHAR2 (10) NOT NULL)
 Storage (INITIAL 5K NEXT 5K
     PCTINCREASE 1)
/

Create Table Ans_Pers (
  Id_bok                              NUMBER(18) NOT NULL,
  P_Id_er                             VARCHAR2 (20) NOT NULL,
  Koden_for                           VARCHAR2 (10) NOT NULL)
 Storage (INITIAL 5K NEXT 5K
     PCTINCREASE 1)
/

Create Table Bibl_konr_nr (
  Id_dok                              NUMBER(18) NOT NULL,
  B_nr_bibl_konr                      VARCHAR2 (20) NOT NULL)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Table Bibl_nr (
  B_nr_for                            VARCHAR2 (20) NOT NULL,
  B_Kilde_nr_kilde                    VARCHAR2 (100) NOT NULL)
 Storage (INITIAL 8K NEXT 8K
     PCTINCREASE 1)
/

Create Table Bokspraak (
```

```
  Id_skrevet_paa                    NUMBER(18) NOT NULL,
  Spraak_brukt_i                    VARCHAR2 (20) NOT NULL)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Table Dok_Eier (
  Id_har                            NUMBER(18) NOT NULL,
  Navnet_eier                       VARCHAR2 (300) NOT NULL,
  Ant_eks                           VARCHAR2 (100) NULL,
  Filial_for                        VARCHAR2 (100) NULL,
  Hylle_plass                       VARCHAR2 (100) NULL)
 Storage (INITIAL 17K NEXT 17K
     PCTINCREASE 1)
/

Create Table Dok_Lev (
  LevId_lev                         VARCHAR2 (100) NOT NULL,
  Id_dok                            NUMBER(18) NOT NULL)
 Storage (INITIAL 7K NEXT 7K
     PCTINCREASE 1)
/

Create Table Dok_Note (
  Id_dok                            NUMBER(18) NOT NULL,
  Noten_note                        VARCHAR2 (500) NOT NULL)
 Storage (INITIAL 20K NEXT 20K
     PCTINCREASE 1)
/

Create Table Dok_kode (
  F_Kode_lokal                      VARCHAR2 (100) NOT NULL,
  Id_dok                            NUMBER(18) NOT NULL)
 Storage (INITIAL 7K NEXT 7K
     PCTINCREASE 1)
/

Create Table Dokument (
  Id_for                            NUMBER(18) NOT NULL,
  Aarstall_slutt_aar                VARCHAR2 (50) NULL,
  Aarstall_start_aar                VARCHAR2 (50) NULL,
  Aarstall_utg_aar                  VARCHAR2 (50) NULL,
  CR_hovednr                        VARCHAR2 (100) NULL,
  Tid_reg_dato                      VARCHAR2 (100) NULL,
  Bokstav_katalogisering            VARCHAR2 (10) NULL,
  L_Navn_utgivelse                  VARCHAR2 (100) NULL,
  Navnet_ansvarlig                  VARCHAR2 (300) NULL,
  Navnet_forkortet_tittel           VARCHAR2 (300) NULL,
  Navnet_nokkel_tittel              VARCHAR2 (300) NULL,
  Navnet_utg_forlag                 VARCHAR2 (300) NULL,
  Navnet_utg_sted                   VARCHAR2 (300) NULL,
  Navnet_utg_trykkeri               VARCHAR2 (300) NULL,
  Navnet_utg_trykksted              VARCHAR2 (300) NULL,
  Lengde_for                        VARCHAR2 (300) NULL,
  Frekvens_utgivelse                VARCHAR2 (100) NULL,
  Statusen_for                      VARCHAR2 (50) NULL,
  Tilfoyelse_nforklaring            VARCHAR2 (300) NULL,
  Utgaven_av                        VARCHAR2 (300) NULL)
 Storage (INITIAL 24K NEXT 24K
     PCTINCREASE 1)
/
```

```
Create Table Emneord (
  Id_bok                          NUMBER(18) NOT NULL,
  Ordet_ord                       VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table ISBN (
  ISBN_nr_for                     VARCHAR2 (15) NOT NULL,
  Betingelsen_leverings           VARCHAR2 (50) NULL,
  Id_feil                         NUMBER(18) NULL,
  Id_riktig                       NUMBER(18) NULL,
  Info_innbinding                 VARCHAR2 (500) NULL,
  Tilfoyelse_tilfoyelser          VARCHAR2 (300) NULL)
 Storage (INITIAL 30K NEXT 30K
     PCTINCREASE 1)
/

Create Table ISSN (
  ISSN_nr_for                     VARCHAR2 (15) NOT NULL,
  Id_feil                         NUMBER(18) NULL,
  Id_riktig                       NUMBER(18) NULL)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Table Konferanse (
  Konf_id_for                     VARCHAR2 (20) NOT NULL,
  Tid_for                         VARCHAR2 (100) NULL,
  Navnet_konf_navn                VARCHAR2 (300) NULL,
  Navnet_stedsnavn                VARCHAR2 (300) NULL,
  Tall_paa                        VARCHAR2 (20) NULL)
 Storage (INITIAL 24K NEXT 24K
     PCTINCREASE 1)
/

Create Table Korporasjon (
  Korp_id_for                     VARCHAR2 (20) NOT NULL,
  Tid_for                         VARCHAR2 (100) NULL,
  Navnet_korp_navn                VARCHAR2 (300) NULL,
  Navnet_stedsnavn                VARCHAR2 (300) NULL,
  Tall_for                        VARCHAR2 (20) NULL)
 Storage (INITIAL 24K NEXT 24K
     PCTINCREASE 1)
/

Create Table Lev_beh (
  Id_dok                          NUMBER(18) NOT NULL,
  Navnet_leverer                  VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table Leverandor (
  LevId_for                       VARCHAR2 (100) NOT NULL,
  Adr_adresse                     VARCHAR2 (100) NULL,
  L_Navn_for                      VARCHAR2 (100) NULL,
  Navnet_lev_navn                 VARCHAR2 (300) NULL,
  Postnr_for                      VARCHAR2 (50) NULL)
 Storage (INITIAL 17K NEXT 17K
```

```
     PCTINCREASE 1)
/

Create Table Lokale_Ord (
  Id_bok                              NUMBER(18) NOT NULL,
  Ordet_ord                           VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table Nokkelord (
  Id_bok                              NUMBER(18) NOT NULL,
  Ordet_ord                           VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table Person (
  P_Id_for                            VARCHAR2 (20) NOT NULL,
  Navnet_paa                          VARCHAR2 (300) NOT NULL,
  Aarstall_for                        VARCHAR2 (50) NULL,
  L_Navn_nasjonalitet                 VARCHAR2 (100) NULL,
  Tilfoyelse_tilfoyelser              VARCHAR2 (300) NULL)
 Storage (INITIAL 24K NEXT 24K
     PCTINCREASE 1)
/

Create Table Tittel (
  T_id_for                            VARCHAR2 (20) NOT NULL,
  Id_dok                              NUMBER(18) NOT NULL,
  Koden_t_kode                        VARCHAR2 (10) NOT NULL,
  Navnet_hovedtittel                  VARCHAR2 (300) NOT NULL,
  Aarstall_aarstall                   VARCHAR2 (50) NULL,
  Aarstall_utgivelsesaar              VARCHAR2 (50) NULL,
  Info_om                             VARCHAR2 (500) NULL)
 Storage (INITIAL 30K NEXT 30K
     PCTINCREASE 1)
/

Create Table URL (
  URL_Adr_for                         VARCHAR2 (200) NOT NULL,
  Storrelse_str                       VARCHAR2 (50) NULL,
  Formatet_for                        VARCHAR2 (100) NULL,
  Noten_ekstern                       VARCHAR2 (500) NULL,
  Noten_intern                        VARCHAR2 (500) NULL)
 Storage (INITIAL 37K NEXT 37K
     PCTINCREASE 1)
/

Create Table URL_Dok (
  Id_dok                              NUMBER(18) NOT NULL,
  URL_Adr_url                         VARCHAR2 (200) NOT NULL)
 Storage (INITIAL 10K NEXT 10K
     PCTINCREASE 1)
/

Rem ***
Rem ***  Index Division for the Data Model
Rem ***

Rem ***
```

```
Rem ***  Constraint Division for the Data Model
Rem ***

Rem ***  Uniqueness Constraints

Alter Table AltNr                              Add Constraint I01_AltNr
Primary Key (
  Id_har,
  CR_alt_nr)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Ans_Konf                           Add Constraint I01_Ans_Konf
Primary Key (
  Id_dok,
  Konf_id_ansvarlig)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Ans_Korp                           Add Constraint I01_Ans_Korp
Primary Key (
  Id_dok,
  Korp_id_ansvarlig)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Ans_Pers                           Add Constraint I01_Ans_Pers
Primary Key (
  Id_bok,
  P_Id_er)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Bibl_konr_nr                       Add Constraint I01_Bibl_konr_nr
Primary Key (
  Id_dok,
  B_nr_bibl_konr)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Bibl_nr                            Add Constraint I01_Bibl_nr
Primary Key (
  B_nr_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Bokspraak                          Add Constraint I01_Bokspraak
Primary Key (
  Id_skrevet_paa,
  Spraak_brukt_i)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/
```

```
Alter Table Dok_Eier                          Add Constraint I01_Dok_Eier
Primary Key (
  Id_har,
  Navnet_eier)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Dok_Lev                           Add Constraint I01_Dok_Lev
Primary Key (
  LevId_lev,
  Id_dok)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Dok_Note                          Add Constraint I01_Dok_Note
Primary Key (
  Id_dok,
  Noten_note)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Dok_kode                          Add Constraint I01_Dok_kode
Primary Key (
  F_Kode_lokal,
  Id_dok)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Dokument                          Add Constraint I01_Dokument
Primary Key (
  Id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Emneord                           Add Constraint I01_Emneord
Primary Key (
  Id_bok,
  Ordet_ord)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table ISBN                              Add Constraint I01_ISBN Primary
Key (
  ISBN_nr_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table ISSN                              Add Constraint I01_ISSN Primary
Key (
  ISSN_nr_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/
```

```
Alter Table Konferanse                          Add Constraint I01_Konferanse
Primary Key (
  Konf_id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Korporasjon                         Add Constraint I01_Korporasjon
Primary Key (
  Korp_id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Lev_beh                             Add Constraint I01_Lev_beh
Primary Key (
  Id_dok,
  Navnet_leverer)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Leverandor                          Add Constraint I01_Leverandor
Primary Key (
  LevId_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Lokale_Ord                          Add Constraint I01_Lokale_Ord
Primary Key (
  Id_bok,
  Ordet_ord)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Nokkelord                           Add Constraint I01_Nokkelord
Primary Key (
  Id_bok,
  Ordet_ord)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Person                              Add Constraint I01_Person
Primary Key (
  P_Id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Tittel                              Add Constraint I01_Tittel
Primary Key (
  T_id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table URL                                 Add Constraint I01_URL Primary
Key (
```

```
  URL_Adr_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table URL_Dok                             Add Constraint I01_URL_Dok
Primary Key (
  Id_dok,
  URL_Adr_url)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Rem ***  Referential Subset Constraints

Alter Table AltNr                               Add Constraint Dokument_AltNr
Foreign Key (
  Id_har)
 References Dokument (
  Id_for)
/

Alter Table Ans_Konf                            Add Constraint
Dokument_Ans_Konf Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Ans_Konf                            Add Constraint
Konferanse_Ans_Konf Foreign Key (
  Konf_id_ansvarlig)
 References Konferanse (
  Konf_id_for)
/

Alter Table Ans_Korp                            Add Constraint
Dokument_Ans_Korp Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Ans_Korp                            Add Constraint
Korporasjon_Ans_Korp Foreign Key (
  Korp_id_ansvarlig)
 References Korporasjon (
  Korp_id_for)
/

Alter Table Ans_Pers                            Add Constraint
Dokument_Ans_Pers Foreign Key (
  Id_bok)
 References Dokument (
  Id_for)
/

Alter Table Ans_Pers                            Add Constraint Person_Ans_Pers
Foreign Key (
  P_Id_er)
 References Person (
```

```
  P_Id_for)
/

Alter Table Bibl_konr_nr                        Add Constraint
Dokument_Bibl_konr_nr Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Bibl_konr_nr                        Add Constraint
Bibl_nr_Bibl_konr_nr Foreign Key (
  B_nr_bibl_konr)
 References Bibl_nr (
  B_nr_for)
/

Alter Table Bokspraak                           Add Constraint
Dokument_Bokspraak Foreign Key (
  Id_skrevet_paa)
 References Dokument (
  Id_for)
/

Alter Table Dok_Eier                            Add Constraint
Dokument_Dok_Eier Foreign Key (
  Id_har)
 References Dokument (
  Id_for)
/

Alter Table Dok_Lev                             Add Constraint
Leverandor_Dok_Lev Foreign Key (
  LevId_lev)
 References Leverandor (
  LevId_for)
/

Alter Table Dok_Lev                             Add Constraint Dokument_Dok_Lev
Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Dok_Note                            Add Constraint
Dokument_Dok_Note Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Dok_kode                            Add Constraint
Dokument_Dok_kode Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Emneord                             Add Constraint Dokument_Emneord
Foreign Key (
```

```
  Id_bok)
 References Dokument (
  Id_for)
/

Alter Table ISBN                              Add Constraint Dokument_ISBN
Foreign Key (
  Id_feil)
 References Dokument (
  Id_for)
/

Alter Table ISBN                              Add Constraint Dokument_ISBN1
Foreign Key (
  Id_riktig)
 References Dokument (
  Id_for)
/

Alter Table ISSN                              Add Constraint Dokument_ISSN
Foreign Key (
  Id_feil)
 References Dokument (
  Id_for)
/

Alter Table ISSN                              Add Constraint Dokument_ISSN1
Foreign Key (
  Id_riktig)
 References Dokument (
  Id_for)
/

Alter Table Lev_beh                           Add Constraint Dokument_Lev_beh
Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Lokale_Ord                        Add Constraint
Dokument_Lokale_Ord Foreign Key (
  Id_bok)
 References Dokument (
  Id_for)
/

Alter Table Nokkelord                         Add Constraint
Dokument_Nokkelord Foreign Key (
  Id_bok)
 References Dokument (
  Id_for)
/

Alter Table Tittel                            Add Constraint Dokument_Tittel
Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/
```

```
Alter Table URL_Dok                              Add Constraint Dokument_URL_Dok
Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table URL_Dok                              Add Constraint URL_URL_Dok
Foreign Key (
  URL_Adr_url)
 References URL (
  URL_Adr_for)
/

Rem ***  Indexes (Duplicate) on Referential Subset

Create Index Dokument_AltNr                                  on AltNr (
  Id_har)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Ans_Konf                               on Ans_Konf (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Konferanse_Ans_Konf                             on Ans_Konf (
  Konf_id_ansvarlig)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Ans_Korp                               on Ans_Korp (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Korporasjon_Ans_Korp                            on Ans_Korp (
  Korp_id_ansvarlig)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Ans_Pers                               on Ans_Pers (
  Id_bok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Person_Ans_Pers                                 on Ans_Pers (
  P_Id_er)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Bibl_konr_nr                           on Bibl_konr_nr (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
```

```
     PCTINCREASE 1)
/

Create Index Bibl_nr_Bibl_konr_nr                       on Bibl_konr_nr (
  B_nr_bibl_konr)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Bokspraak                         on Bokspraak (
  Id_skrevet_paa)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Dok_Eier                          on Dok_Eier (
  Id_har)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Leverandor_Dok_Lev                         on Dok_Lev (
  LevId_lev)
 Storage (INITIAL 7K NEXT 7K
     PCTINCREASE 1)
/

Create Index Dokument_Dok_Lev                           on Dok_Lev (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Dok_Note                          on Dok_Note (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Dok_kode                          on Dok_kode (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Emneord                           on Emneord (
  Id_bok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_ISBN                              on ISBN (
  Id_feil)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_ISBN1                             on ISBN (
  Id_riktig)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
```

```
/

Create Index Dokument_ISSN                                  on ISSN (
  Id_feil)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_ISSN1                                 on ISSN (
  Id_riktig)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Lev_beh                               on Lev_beh (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Lokale_Ord                            on Lokale_Ord (
  Id_bok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Nokkelord                             on Nokkelord (
  Id_bok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Tittel                                on Tittel (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_URL_Dok                               on URL_Dok (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index URL_URL_Dok                                    on URL_Dok (
  URL_Adr_url)
 Storage (INITIAL 10K NEXT 10K
     PCTINCREASE 1)
/

Rem ***  Domain Validations

Rem ***
Rem ***  Function procedures for the Data Model
Rem ***

Rem ***
Rem ***  Packages (with accessor functions) used for reselecting rows...
Rem ***

CREATE OR REPLACE PACKAGE AltNr_Pack AS
```

```
  Id_har AltNr.Id_har%TYPE;
  CR_alt_nr AltNr.CR_alt_nr%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Ans_Kon_Pack AS
  Id_dok Ans_Konf.Id_dok%TYPE;
  Konf_id_ansvarlig Ans_Konf.Konf_id_ansvarlig%TYPE;
  Koden_for Ans_Konf.Koden_for%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Ans_Kor_Pack AS
  Id_dok Ans_Korp.Id_dok%TYPE;
  Korp_id_ansvarlig Ans_Korp.Korp_id_ansvarlig%TYPE;
  Koden_for Ans_Korp.Koden_for%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Ans_Per_Pack AS
  Id_bok Ans_Pers.Id_bok%TYPE;
  P_Id_er Ans_Pers.P_Id_er%TYPE;
  Koden_for Ans_Pers.Koden_for%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Bibl_ko_Pack AS
  Id_dok Bibl_konr_nr.Id_dok%TYPE;
  B_nr_bibl_konr Bibl_konr_nr.B_nr_bibl_konr%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Bibl_nr_Pack AS
  B_nr_for Bibl_nr.B_nr_for%TYPE;
  B_Kilde_nr_kilde Bibl_nr.B_Kilde_nr_kilde%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Bokspra_Pack AS
  Id_skrevet_paa Bokspraak.Id_skrevet_paa%TYPE;
  Spraak_brukt_i Bokspraak.Spraak_brukt_i%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Dok_Eie_Pack AS
  Id_har Dok_Eier.Id_har%TYPE;
  Navnet_eier Dok_Eier.Navnet_eier%TYPE;
  Ant_eks Dok_Eier.Ant_eks%TYPE;
  Filial_for Dok_Eier.Filial_for%TYPE;
  Hylle_plass Dok_Eier.Hylle_plass%TYPE;
  mutating BOOLEAN;
END;
/
```

```
CREATE OR REPLACE PACKAGE Dok_Lev_Pack AS
  LevId_lev Dok_Lev.LevId_lev%TYPE;
  Id_dok Dok_Lev.Id_dok%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Dok_Not_Pack AS
  Id_dok Dok_Note.Id_dok%TYPE;
  Noten_note Dok_Note.Noten_note%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Dok_kod_Pack AS
  F_Kode_lokal Dok_kode.F_Kode_lokal%TYPE;
  Id_dok Dok_kode.Id_dok%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Dokumen_Pack AS
  Id_for Dokument.Id_for%TYPE;
  Aarstall_slutt_aar Dokument.Aarstall_slutt_aar%TYPE;
  Aarstall_start_aar Dokument.Aarstall_start_aar%TYPE;
  Aarstall_utg_aar Dokument.Aarstall_utg_aar%TYPE;
  CR_hovednr Dokument.CR_hovednr%TYPE;
  Tid_reg_dato Dokument.Tid_reg_dato%TYPE;
  Bokstav_katalogisering Dokument.Bokstav_katalogisering%TYPE;
  L_Navn_utgivelse Dokument.L_Navn_utgivelse%TYPE;
  Navnet_ansvarlig Dokument.Navnet_ansvarlig%TYPE;
  Navnet_forkortet_tittel Dokument.Navnet_forkortet_tittel%TYPE;
  Navnet_nokkel_tittel Dokument.Navnet_nokkel_tittel%TYPE;
  Navnet_utg_forlag Dokument.Navnet_utg_forlag%TYPE;
  Navnet_utg_sted Dokument.Navnet_utg_sted%TYPE;
  Navnet_utg_trykkeri Dokument.Navnet_utg_trykkeri%TYPE;
  Navnet_utg_trykksted Dokument.Navnet_utg_trykksted%TYPE;
  Lengde_for Dokument.Lengde_for%TYPE;
  Frekvens_utgivelse Dokument.Frekvens_utgivelse%TYPE;
  Statusen_for Dokument.Statusen_for%TYPE;
  Tilfoyelse_nforklaring Dokument.Tilfoyelse_nforklaring%TYPE;
  Utgaven_av Dokument.Utgaven_av%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Emneord_Pack AS
  Id_bok Emneord.Id_bok%TYPE;
  Ordet_ord Emneord.Ordet_ord%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE ISBN_Pack AS
  ISBN_nr_for ISBN.ISBN_nr_for%TYPE;
  Betingelsen_leverings ISBN.Betingelsen_leverings%TYPE;
  Id_feil ISBN.Id_feil%TYPE;
  Id_riktig ISBN.Id_riktig%TYPE;
  Info_innbinding ISBN.Info_innbinding%TYPE;
  Tilfoyelse_tilfoyelser ISBN.Tilfoyelse_tilfoyelser%TYPE;
  mutating BOOLEAN;
```

```
END;
/

CREATE OR REPLACE PACKAGE ISSN_Pack AS
  ISSN_nr_for ISSN.ISSN_nr_for%TYPE;
  Id_feil ISSN.Id_feil%TYPE;
  Id_riktig ISSN.Id_riktig%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Konfera_Pack AS
  Konf_id_for Konferanse.Konf_id_for%TYPE;
  Tid_for Konferanse.Tid_for%TYPE;
  Navnet_konf_navn Konferanse.Navnet_konf_navn%TYPE;
  Navnet_stedsnavn Konferanse.Navnet_stedsnavn%TYPE;
  Tall_paa Konferanse.Tall_paa%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Korpora_Pack AS
  Korp_id_for Korporasjon.Korp_id_for%TYPE;
  Tid_for Korporasjon.Tid_for%TYPE;
  Navnet_korp_navn Korporasjon.Navnet_korp_navn%TYPE;
  Navnet_stedsnavn Korporasjon.Navnet_stedsnavn%TYPE;
  Tall_for Korporasjon.Tall_for%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Lev_beh_Pack AS
  Id_dok Lev_beh.Id_dok%TYPE;
  Navnet_leverer Lev_beh.Navnet_leverer%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Leveran_Pack AS
  LevId_for Leverandor.LevId_for%TYPE;
  Adr_adresse Leverandor.Adr_adresse%TYPE;
  L_Navn_for Leverandor.L_Navn_for%TYPE;
  Navnet_lev_navn Leverandor.Navnet_lev_navn%TYPE;
  Postnr_for Leverandor.Postnr_for%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Lokale_Pack AS
  Id_bok Lokale_Ord.Id_bok%TYPE;
  Ordet_ord Lokale_Ord.Ordet_ord%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Nokkelo_Pack AS
  Id_bok Nokkelord.Id_bok%TYPE;
  Ordet_ord Nokkelord.Ordet_ord%TYPE;
  mutating BOOLEAN;
END;
/
```

```
CREATE OR REPLACE PACKAGE Person_Pack AS
  P_Id_for Person.P_Id_for%TYPE;
  Navnet_paa Person.Navnet_paa%TYPE;
  Aarstall_for Person.Aarstall_for%TYPE;
  L_Navn_nasjonalitet Person.L_Navn_nasjonalitet%TYPE;
  Tilfoyelse_tilfoyelser Person.Tilfoyelse_tilfoyelser%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Tittel_Pack AS
  T_id_for Tittel.T_id_for%TYPE;
  Id_dok Tittel.Id_dok%TYPE;
  Koden_t_kode Tittel.Koden_t_kode%TYPE;
  Navnet_hovedtittel Tittel.Navnet_hovedtittel%TYPE;
  Aarstall_aarstall Tittel.Aarstall_aarstall%TYPE;
  Aarstall_utgivelsesaar Tittel.Aarstall_utgivelsesaar%TYPE;
  Info_om Tittel.Info_om%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE URL_Pack AS
  URL_Adr_for URL.URL_Adr_for%TYPE;
  Storrelse_str URL.Storrelse_str%TYPE;
  Formatet_for URL.Formatet_for%TYPE;
  Noten_ekstern URL.Noten_ekstern%TYPE;
  Noten_intern URL.Noten_intern%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE URL_Dok_Pack AS
  Id_dok URL_Dok.Id_dok%TYPE;
  URL_Adr_url URL_Dok.URL_Adr_url%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PROCEDURE Init_Packs IS
BEGIN
  AltNr_Pack.mutating := FALSE;
  Ans_Kon_Pack.mutating := FALSE;
  Ans_Kor_Pack.mutating := FALSE;
  Ans_Per_Pack.mutating := FALSE;
  Bibl_ko_Pack.mutating := FALSE;
  Bibl_nr_Pack.mutating := FALSE;
  Bokspra_Pack.mutating := FALSE;
  Dok_Eie_Pack.mutating := FALSE;
  Dok_Lev_Pack.mutating := FALSE;
  Dok_Not_Pack.mutating := FALSE;
  Dok_kod_Pack.mutating := FALSE;
  Dokumen_Pack.mutating := FALSE;
  Emneord_Pack.mutating := FALSE;
  ISBN_Pack.mutating := FALSE;
  ISSN_Pack.mutating := FALSE;
  Konfera_Pack.mutating := FALSE;
  Korpora_Pack.mutating := FALSE;
  Lev_beh_Pack.mutating := FALSE;
  Leveran_Pack.mutating := FALSE;
```

```
  Lokale_Pack.mutating := FALSE;
  Nokkelo_Pack.mutating := FALSE;
  Person_Pack.mutating := FALSE;
  Tittel_Pack.mutating := FALSE;
  URL_Pack.mutating := FALSE;
  URL_Dok_Pack.mutating := FALSE;
END;
/

@Gruppering_userprocs.sql

Rem ***
Rem ***  Table triggers for the Data Model
Rem ***

Rem ***  Delete triggers

Rem ***  Insert/update triggers



Rem ***  End Data Model
```

D Javaprogram for den andre relasjonelle databasen

```
1  import java.sql.*;
2  import oracle.sql.*;
3  import oracle.jdbc.driver.*;
4  import easyIO.*;
5  import java.util.*;
6  import java.io.*;
7
8  public class lese_norart {
9      //Deklarasjoner
10     private In inn;
11     private HashMap bok = null;
12     private Connection conn;
13     private int personid = 52245;
14     private int korporasjonid = 3279;
15     private int tittelid = 0;
16     private int ny_levid = 0;
17     private int ant_art = 0;
18
19     public lese_norart(String filnavn) throws SQLException{
20
21         // Last ned Oracle JDBC driveren
22         DriverManager.registerDriver(new
23 oracle.jdbc.driver.OracleDriver());
24
25         //Koble til Oracle
26         kobleTil();
27
28         //Lese dokumenter fra fila
29         inn = new In(filnavn);
30         lesFraFil();
31
32         System.exit(0);
33
34     }//end konstruktør lese
35
36     public void kobleTil() {
37         In tastatur = new In();
38         String brukernavn;
39         String passord;
40
41         System.out.print("Brukernavn: ");
42         brukernavn = tastatur.inString();
43         System.out.print("Passord: ");
44         passord = tastatur.inString();
45
46         try {
47             conn = DriverManager.getConnection(
48 "jdbc:oracle:thin:@blot.ifi.uio.no:1521:IFIFORSK",
49 brukernavn, passord);
50         }
51         catch(Exception e) {
```

```
52              System.err.println("Feil: " + e);
53              System.exit(1);
54          }
55
56      }//end kobleTil
57
58
59      public void lesFraFil() throws SQLException {
60
61          char [] kode = new char[3];
62          char delkode;
63          int helekoden;
64          int count = 0;
65          String tegn;
66
67          tegn = inn.inWord();
68          while(!tegn.startsWith("*") && !tegn.startsWith("^")){
69              tegn = inn.inWord();
70          }
71
72          while(!inn.lastItem()){
73              //Når metodene er ferdig returnerer de ordet de er på.
74              //Ordet er da ^ eller *xxx.
75
76              //Hvis ordet begynner med ^, les videre til koden.
77              if(tegn.equals("^")) {
78                  ant_art++;
79                  //Da har vi en ny bok
80                  if(count > 0) {
81                      //Ferdig med en bok. Den må da legges
82                      //inn i Oracle
83                      leggInn(bok);
84                      //HashMapen nulles ut
85                      bok = new HashMap();
86
87                      //Nuller ut ArrayListene
88                      ArrayList nokkelordene = new ArrayList();
89                      bok.put("nokkelordene", nokkelordene);
90                      ArrayList titlene = new ArrayList();
91                      bok.put("titlene", titlene);
92                      ArrayList personene = new ArrayList();
93                      bok.put("personene", personene);
94                      ArrayList korpene = new ArrayList();
95                      bok.put("korpene", korpene);
96                      ArrayList deweykoder = new ArrayList();
97                      bok.put("deweykoder", deweykoder);
98                      ArrayList sammendragene = new ArrayList();
99                      bok.put("sammendragene", sammendragene);
100
101                 }//end if
102                 else {
103                     //Første boka, oppretter da en HashMap bok
104                     bok = new HashMap();
105
```

```
106                     //Opprett ArrayLister der
107                     //metoden kalles flere ganger
108                     ArrayList nokkelordene = new ArrayList();
109                     bok.put("nokkelordene", nokkelordene);
110                     ArrayList titlene = new ArrayList();
111                     bok.put("titlene", titlene);
112                     ArrayList personene = new ArrayList();
113                     bok.put("personene", personene);
114                     ArrayList korpene = new ArrayList();
115                     bok.put("korpene", korpene);
116                     ArrayList deweykoder = new ArrayList();
117                     bok.put("deweykoder", deweykoder);
118                     ArrayList sammendragene = new ArrayList();
119                     bok.put("sammendragene", sammendragene);
120
121                 }//end else
122                 count++;
123                 tegn = inn.inWord();
124
125             }//end if ^
126
127             //Hvis ordet begynner med *, så har vi koden.
128             if(tegn.startsWith("*")){
129                 //Finner ny kode
130                 helekoden = new Integer(tegn.substring(1, 4)).
131                     intValue();
132
133                 switch(helekoden) {
134
135                 case 1: tegn = id(tegn);
136                     break;
137                 case 8: tegn = infokoder(tegn);
138                     break;
139                 case 41: tegn = spraak();
140                     break;
141                 case 82: tegn = dewey_kode(tegn);
142                     break;
143                 case 100: tegn = person(helekoden);
144                     break;
145                 case 110: tegn = korporasjon(helekoden);
146                     break;
147                 case 245: tegn = tittel();
148                     break;
149                 case 300: tegn = omfang();
150                     break;
151                 case 520: tegn = sammendrag();
152                     break;
153                 case 600: tegn = person(helekoden);
154                     break;
155                 case 610: tegn = korporasjon(helekoden);
156                     break;
157                 case 653: tegn = nokkelord();
158                     break;
159                 case 700: tegn = person(helekoden);
```

```
160                     break;
161                 case 773: tegn = vertsdokument();
162                     break;
163                 default: tegn = videre();
164                     break;
165                 }//end switch
166 
167             }//end if*
168 
169         }//end while
170 
171         //legg inn siste bok
172         leggInn(bok);
173         inn.close();
174 
175     }//end lesFraFil()
176 
177 
178     public void leggInn(HashMap h) throws SQLException{
179 
180         System.out.println("*************************");
181         System.out.println("id: " + h.get("id"));
182         System.out.println("*************************");
183 
184         // Lag et SQL-setningsobjekt
185         Statement stmt = conn.createStatement();
186 
187         //Legger attributter inn i tabellen Dokument i Oracle.
188         //*******************************************************
189 
190         try {
191 
192             stmt.executeUpdate("INSERT INTO Dokument(Id_for,
193 Aarstall_vert, Tid_reg_dato, ISSN_nr_issn, Info_tittel,
194 Navnet_tittel, Lengde_for, Rel_nr_nr, Tittelen_paa) VALUES("
195 + h.get("id") + ",'" + h.get("aarstall_vert") + "','" +
196 h.get("tid_reg_dato") + "','" + h.get("issn_nr") + "','" +
197 h.get("tittel_info") + "','" + h.get("navnet_tittel") + "','"
198 + h.get("omfang") + "','" + h.get("rel_nr") + "','"
199 + h.get("hovedtittel") + "')");
200 
201 
202         //Legger attributter inn i tabellen Bokspraak i Oracle.
203         //*******************************************************
204 
205         ArrayList sp = (ArrayList)h.get("spraakene");
206         if(sp != null) {
207             for(int i = 0; i < sp.size(); i++) {
208                 //Må sjekke at ikke samme språk ligger i
209                 //bokspraak og spraakene.
210                 if(!sp.get(i).equals(h.get("bokspraak"))) {
211                     stmt.executeUpdate("INSERT INTO Bokspraak(
212 Spraak_brukt_i, Id_skrevet_paa) VALUES('" + sp.get(i) + "',"
213 + h.get("id") + ")");
```

```
214                 }//end if
215             }//end for
216 
217         }//end if
218         if(h.get("bokspraak") != null) {
219             stmt.executeUpdate("INSERT INTO Bokspraak(Spraak_brukt_i,
220 Id_skrevet_paa) VALUES('" + h.get("bokspraak") + "'," +
221 h.get("id") + ")");
222         }//end if
223 
224 
225         //Legger attributter inn i tabellen Dok_Dew i Oracle.
226         //*********************************************************
227 
228         ArrayList de = (ArrayList)h.get("deweykoder");
229         for(int i = 0; i < de.size(); i++) {
230             //System.out.println("Dewey: " + de.get(i));
231             stmt.executeUpdate("INSERT INTO Dok_Dew(Id_dok,
232 Deweykode_dew_kode) VALUES(" + h.get("id") + "," + "'"
233 + de.get(i) + "'" + ")");
234         }//end for
235 
236 
237         //Legger attributter inn i tabellene Person og Ans_Pers.
238         //*********************************************************
239 
240         ArrayList lpers = (ArrayList)h.get("personene");
241 
242         for(int i = 0; i < lpers.size(); i++) {
243             String[] spers = (String[])lpers.get(i);
244 
245             if(spers[1] != null) {
246                 stmt.executeUpdate("INSERT INTO Person(P_Id_for,
247 Navnet_paa) VALUES('" + spers[1] + "','" + spers[2] + "'" + ")");
248 
249                 stmt.executeUpdate("INSERT INTO Ans_Pers(Id_bok,
250 P_Id_er, Koden_for) VALUES(" + h.get("id") + "," + "'" +
251 spers[1] + "','" + spers[0] + "'" + ")");
252 
253             }//end if
254         }//end for
255 
256 
257         //Legger inn i tabellene Korporasjon og Ans_Korp i Oracle.
258         //*********************************************************
259 
260         ArrayList lkorp = (ArrayList)h.get("korpene");
261 
262         for(int i = 0; i < lkorp.size(); i++) {
263             String[] skorp = (String[])lkorp.get(i);
264 
265             if(!skorp[1].equals(null)) {
266                 stmt.executeUpdate("INSERT INTO Korporasjon(
267 Korp_id_for, Navnet_korp_navn) VALUES('" + skorp[1] + "','" +
```

```
skorp[2] + "'" + ")");

                stmt.executeUpdate("INSERT INTO Ans_Korp(Id_dok,
Korp_id_ansvarlig, Koden_for) VALUES(" + h.get("id") + "," + "'"
+ skorp[1] + "','" + skorp[0] + "'" + ")");

            }//end if

        }//end for


        //Legger attributter inn i tabellen Sammendrag i Oracle.
        //*******************************************************

        ArrayList sd = (ArrayList)h.get("sammendragene");
        for(int i = 0; i < sd.size(); i++) {
            stmt.executeUpdate("INSERT INTO Sammendrag(Id_dok,
Teksten_sammendrag) VALUES(" + h.get("id") + "," + "'"
+ sd.get(i) + "'" + ")");
        }//end for

        //Legger attributter inn i tabellen Nokkelord i Oracle.
        //*******************************************************

        ArrayList no = (ArrayList)h.get("nokkelordene");
        for(int i = 0; i < no.size(); i++) {
            stmt.executeUpdate("INSERT INTO Nokkelord(Id_bok,
Ordet_ord) VALUES(" + h.get("id") + "," + "'" + no.get(i)
+ "'" + ")");
        }//end for

        stmt.close();

        }catch(Exception e) {
            System.err.println(e);
            System.exit(1);
        }


    }//end leggInn()


    public String videre() {
        //Lese videre til neste kode
        String vid = null;

        vid = inn.inWord();
        while(!vid.startsWith("*") && !vid.startsWith("^")) {
            vid = inn.inWord();
        }//end while

        return vid;

    }//end videre()
```

```
322
323
324     public String id(String t) {
325         //Lese dokumentId fra fila og legge den i HashMap bok
326         String id = null;
327         String nest = null;
328
329         id = t.substring(4);
330         bok.put("id",id);
331
332         nest = inn.inWord();
333         while(!nest.startsWith("*") && !nest.startsWith("^")) {
334             nest = inn.inWord();
335         }//end while
336
337         return nest;
338
339     }//end id()
340
341
342     public String infokoder(String st) {
343         //Lese informasjonskoder fra fila og legge inn i bok
344         //00-05 er tid_reg_dato, 35-37 er språk,
345
346         String tid_reg_dato = null;
347         String bokspraak= null;
348         String enn = null;
349         String linja = null;
350         int lengde = 0;
351         int igjen = 0;
352         String nest = null;
353
354         nest = st.substring(4);
355
356         lengde = nest.length();
357         igjen = 38 - lengde;
358         linja = nest + inn.inChar();
359         for(int i = 0; i < igjen; i++) {
360             linja = linja + inn.inChar();
361                 }
362         tid_reg_dato = linja.substring(0,6);
363         if(tid_reg_dato.equals("      ")) {
364             tid_reg_dato = null;
365         }
366         bok.put("tid_reg_dato",tid_reg_dato);
367
368         bokspraak = linja.substring(35,38);
369         if(!bokspraak.equals("mul") && !bokspraak.equals("   ")) {
370             bok.put("bokspraak", bokspraak);
371         }
372
373         enn = inn.inWord();
374         while(!enn.startsWith("*") && !enn.startsWith("^")) {
375             enn = inn.inWord();
```

```
376         }//end while
377
378         return enn;
379     }//end infokoder
380
381
382     public String spraak() {
383         //Lese inn språk og legge det i bok
384
385         String spraak = null;
386         String og = null;
387         String spraaket = null;
388         ArrayList spraakene = new ArrayList();
389
390         og = inn.inWord();
391         while(og.startsWith("$")) {
392             if(og.startsWith("$a")) {
393                 spraak = og.substring(2);
394                 for(int i = 0; i < 30; i+=3) {
395                     if(i+3 <= spraak.length()) {
396                         spraaket = spraak.substring(i,i+3);
397                         if(!spraakene.contains(spraaket)) {
398                             spraakene.add(spraaket);
399                         }//end if
400                     }//end if
401                 }//end for
402                 og = inn.inWord();
403             }//end $a
404             else {
405                 og = inn.inWord();
406                 while(!og.startsWith("$") && !og.startsWith("*")
407                       && !og.startsWith("^")) {
408                     og = inn.inWord();
409                 }//end while
410             }//end else
411         }//end while
412         bok.put("spraakene", spraakene);
413         return og;
414     }//end spraak()
415
416
417     public String dewey_kode(String dk) {
418         String dewey = null;
419         String more = null;
420
421         dewey = dk.substring(8);
422         ArrayList dew = (ArrayList)bok.get("deweykoder");
423         dew.add(dewey);
424         more = inn.inWord();
425         while(!more.startsWith("$") && !more.startsWith("*")
426               && !more.startsWith("^")) {
427             more = inn.inWord();
428         }//end while
429         return more;
```

```
430     }//end dewey_kode()
431
432
433     public String person(int p_kode) {
434         //Lese kode, id og navn for person og legge inn i bok
435
436         String pers_her = null;
437         String pers_mer = null;
438         String pers_id = null;
439         String pers_navn = null;
440         String pers_kode = Integer.toString(p_kode);
441         String pers[] = new String[3];
442
443         pers[0] = pers_kode;
444
445         personid++;
446         pers[1] = Integer.toString(personid);
447
448         pers_her = inn.inWord();
449         while(pers_her.startsWith("$")) {
450             if(pers_her.startsWith("$a")) {
451                 pers_navn = pers_her.substring(2);
452                 pers_mer = inn.inWord();
453                 while(!pers_mer.startsWith("$")
454                       && !pers_mer.startsWith("*")
455                       && !pers_mer.startsWith("^")) {
456                     pers_navn = pers_navn + " " + pers_mer;
457                     pers_mer = inn.inWord();
458                 }
459                 pers[2] = pers_navn;
460                 pers_her = pers_mer;
461             }//end $a
462             else {
463                 pers_her = inn.inWord();
464                 while(!pers_her.startsWith("$")
465                       && !pers_her.startsWith("*")
466                       && !pers_her.startsWith("^")) {
467                     pers_her = inn.inWord();
468                 }//end while
469             }//end else
470         }//end while
471
472         ArrayList l = (ArrayList)bok.get("personene");
473         l.add(pers);
474
475         return pers_her;
476     }//end person()
477
478
479
480     public String korporasjon(int korp_kode) {
481         //Lese inn kode, id og navn for korporasjon
482
483         String korp_her = null;
```

```
484         String korp_mer = null;
485         String korp_id = null;
486         String korp_navn = null;
487         String korporasjon_kode = Integer.toString(korp_kode);
488         String korp[] = new String[3];
489 
490         korp[0] = korporasjon_kode;
491 
492         korporasjonid++;
493         korp[1] = Integer.toString(korporasjonid);
494 
495         korp_her = inn.inWord();
496         while(korp_her.startsWith("$")) {
497             if(korp_her.startsWith("$a")) {
498                 korp_navn = korp_her.substring(2);
499                 korp_mer = inn.inWord();
500                 while(!korp_mer.startsWith("$")
501                       && !korp_mer.startsWith("*")
502                       && !korp_mer.startsWith("^")) {
503                     korp_navn = korp_navn + " " + korp_mer;
504                     korp_mer = inn.inWord();
505                 }
506                 korp[2] = korp_navn;
507                 korp_her = korp_mer;
508             }//end $a
509             else {
510                 korp_her = inn.inWord();
511                 while(!korp_her.startsWith("$")
512                       && !korp_her.startsWith("*")
513                       && !korp_her.startsWith("^")) {
514                     korp_her = inn.inWord();
515                 }//end while
516             }//end else
517         }//end while
518 
519         ArrayList l = (ArrayList)bok.get("korpene");
520         l.add(korp);
521 
522         return korp_her;
523     }//end korporasjon()
524 
525 
526     public String tittel() {
527         //Lese tittel og tittelinformasjon
528         String hovedtittel = null;
529         String tittel_info = null;
530         String a = null;
531         String b = null;
532 
533         a = inn.inWord();
534         while(a.startsWith("$")) {
535             if(a.startsWith("$a")) {
536                 hovedtittel = a.substring(2);
537                 b = inn.inWord();
```

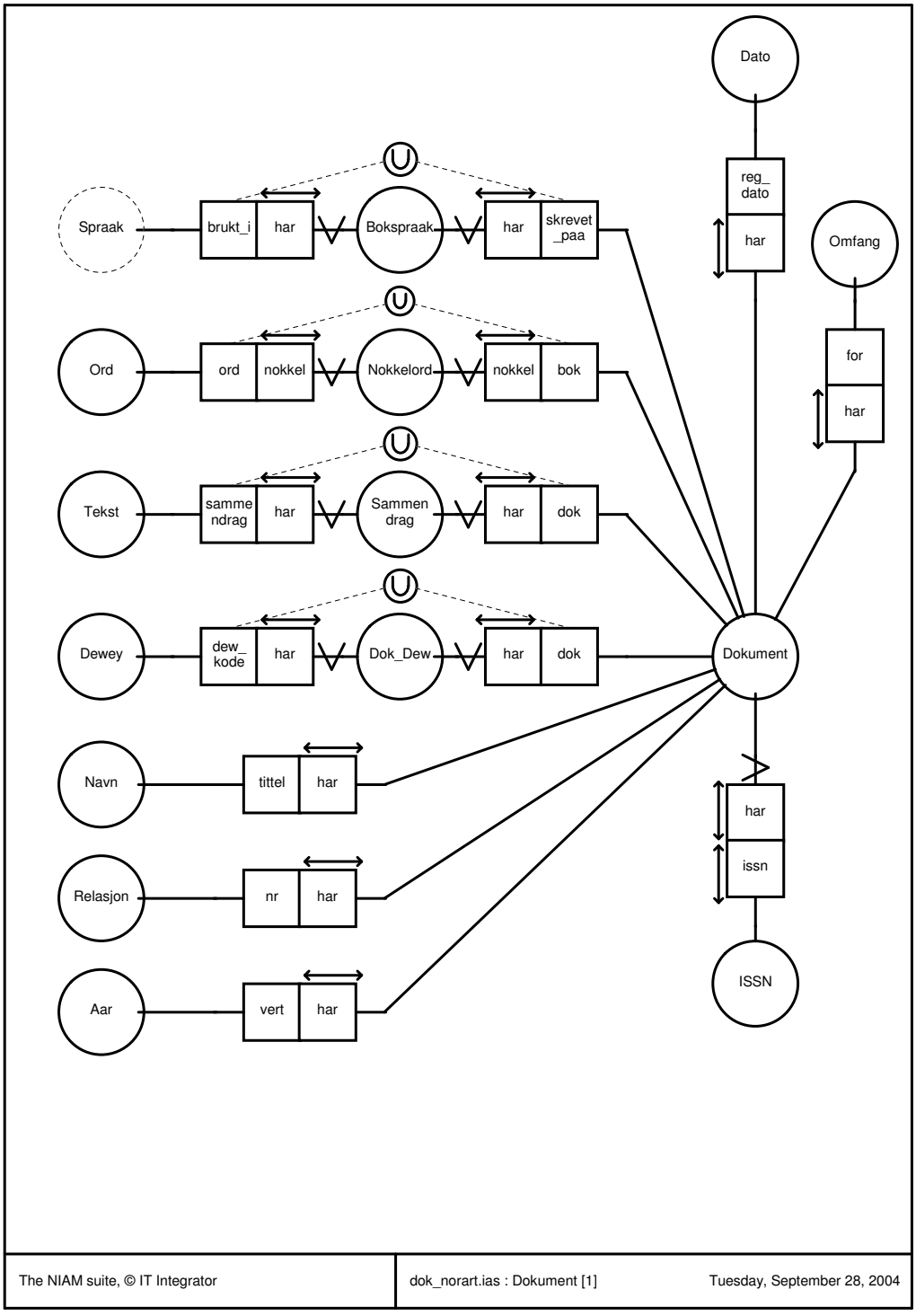
```
538                 while(!b.startsWith("$") && !b.startsWith("*")
539                       && !b.startsWith("^")) {
540                     hovedtittel = hovedtittel + " " + b;
541                     b = inn.inWord();
542                 }
543                 bok.put("hovedtittel",hovedtittel);
544                 a = b;
545             }//end $a
546             else if(a.startsWith("$b")) {
547                 tittel_info = a.substring(2);
548                 b = inn.inWord();
549                 while(!b.startsWith("$") && !b.startsWith("*")
550                       && !b.startsWith("^")) {
551                     tittel_info = tittel_info + " " + b;
552                     b = inn.inWord();
553                 }
554                 bok.put("tittel_info",tittel_info);
555                 a = b;
556             }//end $b
557             else {
558                 a = inn.inWord();
559                 while(!a.startsWith("$") && !a.startsWith("*")
560                       && !a.startsWith("^")) {
561                     a = inn.inWord();
562                 }//end while
563             }//end else
564         }//end while()
565 
566         return a;
567 
568     }//end tittel()
569 
570 
571 
572     public String omfang() {
573         //lese omfang og legge inn i bok.
574 
575         String aa = null;
576         String bb = null;
577         String omfang = null;
578 
579         aa = inn.inWord();
580         while(aa.startsWith("$")) {
581             if(aa.startsWith("$a")) {
582                 omfang = aa.substring(2);
583                 bb = inn.inWord();
584                 while(!bb.startsWith("$") && !bb.startsWith("*")
585                       && !bb.startsWith("^")) {
586                     omfang = omfang + " " + bb;
587                     bb = inn.inWord();
588                 }
589                 bok.put("omfang", omfang);
590                 aa = bb;
591             }//end $a
```

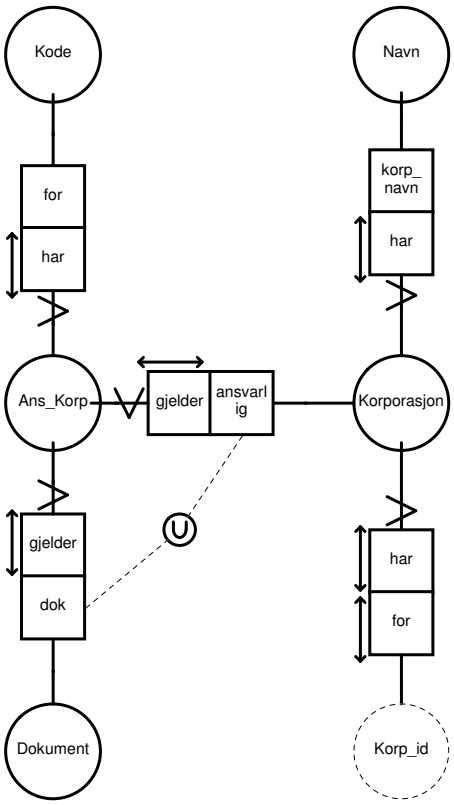
```
592             else {
593                 aa = inn.inWord();
594                 while(!aa.startsWith("$") && !aa.startsWith("*")
595                       && !aa.startsWith("^")) {
596                     aa = inn.inWord();
597                 }//end while
598             }//end else
599         }//end while()
600         return aa;
601     }//end omfang()
602 
603 
604     public String sammendrag() {
605         //Lese sammendrag og legge inn i bok.
606 
607         String uu = null;
608         String pp = null;
609         String sammendrag = null;
610 
611         uu = inn.inWord();
612         while(uu.startsWith("$")) {
613             if(uu.startsWith("$a")) {
614                 sammendrag = uu.substring(2);
615                 pp = inn.inWord();
616                 while(!pp.startsWith("$") && !pp.startsWith("*")
617                       && !pp.startsWith("^")) {
618                     sammendrag = sammendrag + " " + pp;
619                     pp = inn.inWord();
620                 }
621 
622                 ArrayList l = (ArrayList)bok.get("sammendragene");
623                 l.add(sammendrag);
624 
625                 uu = pp;
626             }//end $a
627             else {
628                 uu = inn.inWord();
629                 while(!uu.startsWith("$") && !uu.startsWith("*")
630                       && !uu.startsWith("^")) {
631                     uu = inn.inWord();
632                 }//end while
633             }//end else
634         }//end while()
635 
636         return uu;
637 
638     }//end sammendrag()
639 
640 
641 
642     public String nokkelord() {
643         //Lese nokkelord
644         String nokkelord = null;
645         String ordet = null;
```

```
646         String nestemann = null;
647
648         ordet = inn.inWord();
649         while(ordet.startsWith("$")) {
650             if(ordet.startsWith("$a")) {
651                 nokkelord = ordet.substring(2);
652                 ArrayList l = (ArrayList)bok.get("nokkelordene");
653                 if(!l.contains(nokkelord)) {
654                     l.add(nokkelord);
655                 }//end if
656
657                 nestemann = inn.inWord();
658                 while(!nestemann.startsWith("$")
659                       && !nestemann.startsWith("*")
660                       && !nestemann.startsWith("^")) {
661                     nokkelord = nestemann;
662                     if(!l.contains(nokkelord)) {
663                         l.add(nokkelord);
664                     }//end if
665                     nestemann = inn.inWord();
666                 }
667                 ordet = nestemann;
668             }//end $a
669             else {
670                 ordet = inn.inWord();
671                 while(!ordet.startsWith("$")
672                       && !ordet.startsWith("*")
673                       && !ordet.startsWith("^")) {
674                     ordet = inn.inWord();
675                 }//end while
676             }//end else
677         }//end while
678         return ordet;
679     }//end nokkelord()
680
681
682     public String vertsdokument() {
683         //Lese inn aarstall_vert, issn, vertstittel, relasjonsnr.
684
685         String hele = null;
686         String halve = null;
687         String aarstall_vert = null; //$i
688         String issn_nr = null; //$x
689         String navnet_tittel = null; //$t
690         String rel_nr = null; //$g
691
692         hele = inn.inWord();
693         halve = inn.inWord();
694         while(!halve.startsWith("*") && !halve.startsWith("^")) {
695             hele = hele + " " + halve;
696             halve = inn.inWord();
697         }//end while
698
699         StringTokenizer st = new StringTokenizer(hele,"$");
```

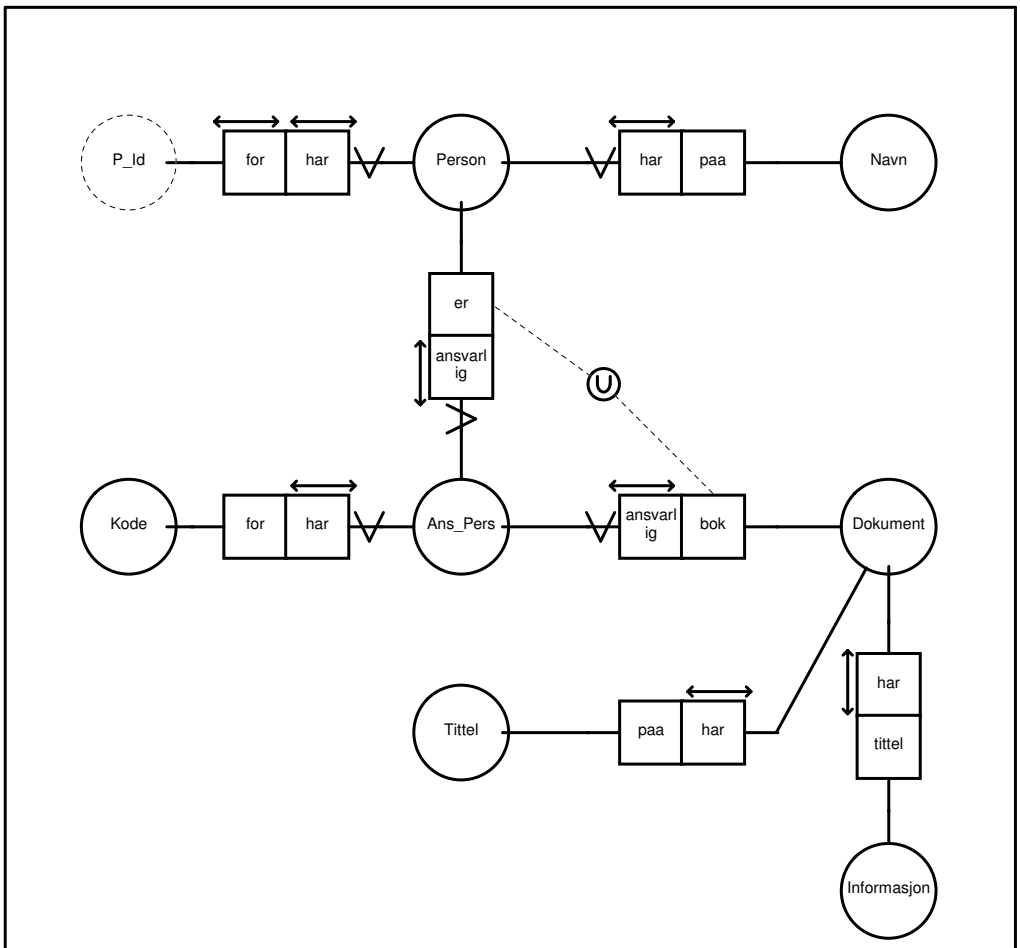
```
700
701         while(st.hasMoreTokens()) {
702
703             String token = st.nextToken();
704
705             if(token.substring(0,1).equals("i")) {
706                 aarstall_vert = token.substring(1);
707                 bok.put("aarstall_vert",aarstall_vert);
708             }//end if $i
709
710             if(token.substring(0,1).equals("x")) {
711                 issn_nr = token.substring(1);
712                 bok.put("issn_nr",issn_nr);
713             }//end if $x
714
715             if(token.substring(0,1).equals("t")) {
716                 navnet_tittel = token.substring(1);
717                 bok.put("navnet_tittel",navnet_tittel);
718             }//end if $t
719
720             if(token.substring(0,1).equals("g")) {
721                 rel_nr = token.substring(1);
722                 bok.put("rel_nr",rel_nr);
723             }//end if $g
724
725         }//end while
726
727         return halve;
728
729     }//end vertsdokument()
730
731
732
733     public static void main( String argv[] ) throws Exception {
734
735         if(argv.length == 1){
736             new lese_norart(argv[0]);
737         }
738         else {
739             System.out.println("Bruk:\n\t java lese 'filnavn'");
740         }
741
742     }//end main
743
744 }//end class lese
```

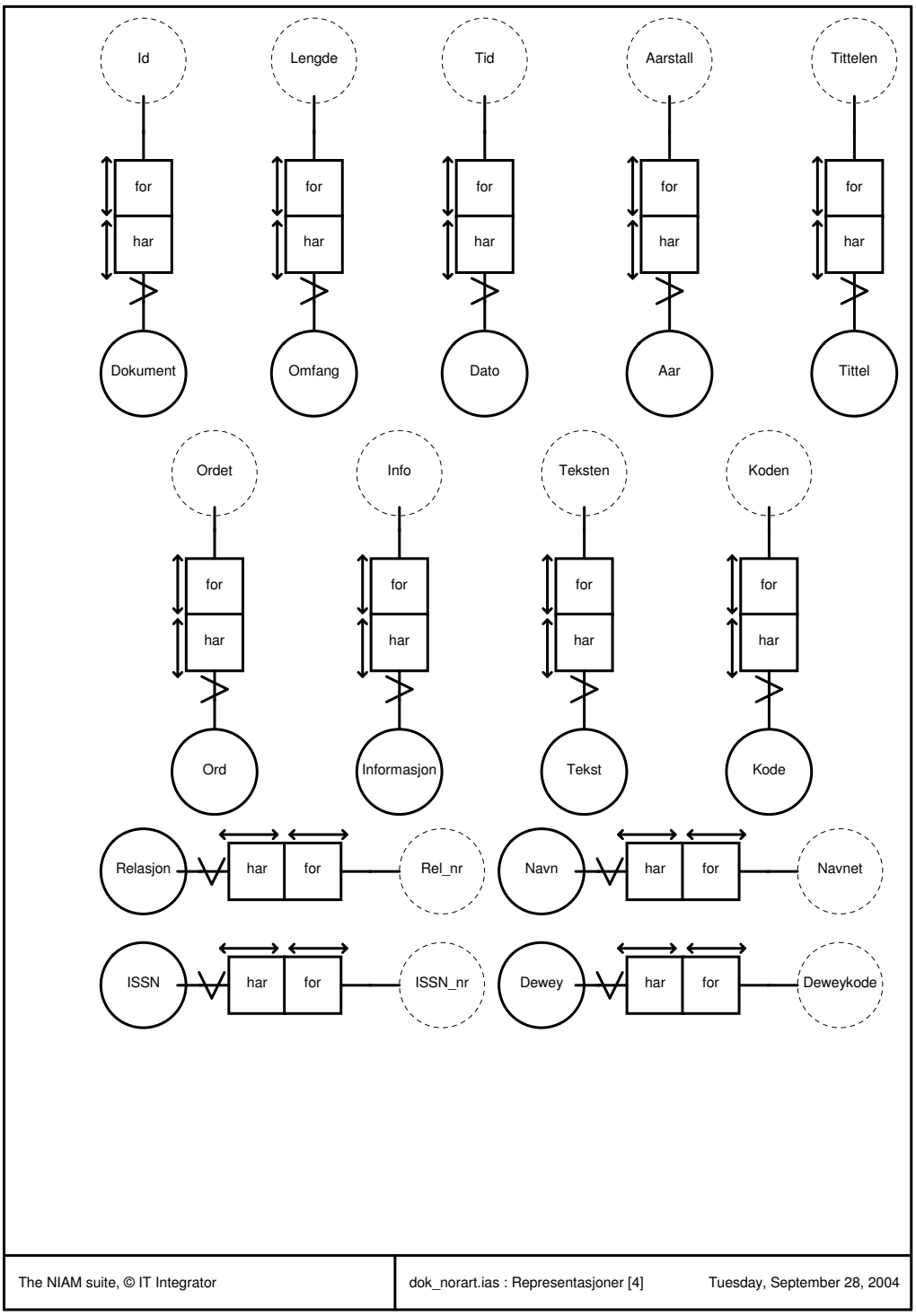
E NIAM-modell for den andre relasjonelle databasen





The NIAM suite, © IT Integrator | dok_norart.ias : Korporasjon [2] | Tuesday, September 28, 2004


P_Id
for
har
Person
har
paa
Navn
er
ansvarl
ig
U
Kode
for
har
Ans_Pers
ansvarl
ig
bok
Dokument
har
tittel
Tittel
paa
har
Informasjon
The NIAM suite, © IT Integrator
dok_norart.ias : Person [3]
Tuesday, September 28, 2004


Id
Lengde
Tid
Aarstall
Tittelen
for
har
Dokument
Omfang
Dato
Aar
Tittel
Ordet
Info
Teksten
Koden
Ord
Informasjon
Tekst
Kode
Relasjon
Rel_nr
Navn
Navnet
ISSN
ISSN_nr
Dewey
Deweykode
The NIAM suite, © IT Integrator
dok_norart.ias : Representasjoner [4]
Tuesday, September 28, 2004

F Oracleskjema for den andre relasjonelle databasen

```
Rem ****************************************************************
Rem
Rem Oracle 7 Database Description File for gruppering
Rem Generation Date is 2004/8/30
Rem Model Number is 10
Rem
Rem (The NIAM Suite 3.4, (c)COPYRIGHT IT Integrator A/S. ALL RIGHTS RESERVED)
Rem
Rem (Gene date 08.09.00)
Rem ****************************************************************

Rem ***
Rem ***  Table Division for the Data Model
Rem ***

Create Table Ans_Korp (
  Id_dok                          NUMBER(18) NOT NULL,
  Korp_id_ansvarlig               VARCHAR2 (20) NOT NULL,
  Koden_for                       VARCHAR2 (10) NOT NULL)
 Storage (INITIAL 5K NEXT 5K
     PCTINCREASE 1)
/

Create Table Ans_Pers (
  Id_bok                          NUMBER(18) NOT NULL,
  P_Id_er                         VARCHAR2 (20) NOT NULL,
  Koden_for                       VARCHAR2 (10) NOT NULL)
 Storage (INITIAL 5K NEXT 5K
     PCTINCREASE 1)
/

Create Table Bokspraak (
  Spraak_brukt_i                  VARCHAR2 (20) NOT NULL,
  Id_skrevet_paa                  NUMBER(18) NOT NULL)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Table Dok_Dew (
  Id_dok                          NUMBER(18) NOT NULL,
  Deweykode_dew_kode              VARCHAR2 (50) NOT NULL)
 Storage (INITIAL 5K NEXT 5K
     PCTINCREASE 1)
/

Create Table Dokument (
  Id_for                          NUMBER(18) NOT NULL,
  ISSN_nr_issn                    VARCHAR2 (50) NOT NULL,
  Aarstall_vert                   VARCHAR2 (50) NULL,
  Tid_reg_dato                    VARCHAR2 (100) NULL,
  Info_tittel                     VARCHAR2 (500) NULL,
  Navnet_tittel                   VARCHAR2 (300) NULL,
  Lengde_for                      VARCHAR2 (300) NULL,
  Rel_nr_nr                       VARCHAR2 (50) NULL,
  Tittelen_paa                    VARCHAR2 (300) NULL)
 Storage (INITIAL 30K NEXT 30K
     PCTINCREASE 1)
/

Create Table Korporasjon (
  Korp_id_for                     VARCHAR2 (20) NOT NULL,
```

```
  Navnet_korp_navn                    VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table Nokkelord (
  Id_bok                              NUMBER(18) NOT NULL,
  Ordet_ord                           VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table Person (
  P_Id_for                            VARCHAR2 (20) NOT NULL,
  Navnet_paa                          VARCHAR2 (300) NOT NULL)
 Storage (INITIAL 14K NEXT 14K
     PCTINCREASE 1)
/

Create Table Sammendrag (
  Id_dok                              NUMBER(18) NOT NULL,
  Teksten_sammendrag                  VARCHAR2 (500) NOT NULL)
 Storage (INITIAL 20K NEXT 20K
     PCTINCREASE 1)
/

Rem ***
Rem ***  Index Division for the Data Model
Rem ***

Rem ***
Rem ***  Constraint Division for the Data Model
Rem ***

Rem ***  Uniqueness Constraints

Alter Table Ans_Korp                          Add Constraint I01_Ans_Korp
Primary Key (
  Id_dok,
  Korp_id_ansvarlig)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Ans_Pers                          Add Constraint I01_Ans_Pers
Primary Key (
  Id_bok,
  P_Id_er)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Bokspraak                         Add Constraint I01_Bokspraak
Primary Key (
  Spraak_brukt_i,
  Id_skrevet_paa)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/
```

```
Alter Table Dok_Dew                             Add Constraint I01_Dok_Dew
Primary Key (
  Id_dok,
  Deweykode_dew_kode)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Dokument                            Add Constraint I01_Dokument
Primary Key (
  Id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Dokument                            Add Constraint I2_Dokument
Unique (
  ISSN_nr_issn)
    Using Index Storage
    (INITIAL 4K NEXT 4K  PCTINCREASE 1)
/

Alter Table Korporasjon                         Add Constraint I01_Korporasjon
Primary Key (
  Korp_id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Nokkelord                           Add Constraint I01_Nokkelord
Primary Key (
  Id_bok,
  Ordet_ord)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Person                              Add Constraint I01_Person
Primary Key (
  P_Id_for)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Alter Table Sammendrag                          Add Constraint I01_Sammendrag
Primary Key (
  Id_dok,
  Teksten_sammendrag)
    Using Index Storage
    (INITIAL 4K NEXT 4K PCTINCREASE 1)
/

Rem ***  Referential Subset Constraints

Alter Table Ans_Korp                            Add Constraint
Dokument_Ans_Korp Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/
```

```
Alter Table Ans_Korp                            Add Constraint
Korporasjon_Ans_Korp Foreign Key (
  Korp_id_ansvarlig)
 References Korporasjon (
  Korp_id_for)
/

Alter Table Ans_Pers                            Add Constraint
Dokument_Ans_Pers Foreign Key (
  Id_bok)
 References Dokument (
  Id_for)
/

Alter Table Ans_Pers                            Add Constraint Person_Ans_Pers
Foreign Key (
  P_Id_er)
 References Person (
  P_Id_for)
/

Alter Table Bokspraak                           Add Constraint
Dokument_Bokspraak Foreign Key (
  Id_skrevet_paa)
 References Dokument (
  Id_for)
/

Alter Table Dok_Dew                             Add Constraint Dokument_Dok_Dew
Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Alter Table Nokkelord                           Add Constraint
Dokument_Nokkelord Foreign Key (
  Id_bok)
 References Dokument (
  Id_for)
/

Alter Table Sammendrag                          Add Constraint
Dokument_Sammendrag Foreign Key (
  Id_dok)
 References Dokument (
  Id_for)
/

Rem ***  Indexes (Duplicate) on Referential Subset

Create Index Dokument_Ans_Korp                            on Ans_Korp (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Korporasjon_Ans_Korp                         on Ans_Korp (
  Korp_id_ansvarlig)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
```

```
/

Create Index Dokument_Ans_Pers                              on Ans_Pers (
  Id_bok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Person_Ans_Pers                                on Ans_Pers (
  P_Id_er)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Bokspraak                             on Bokspraak (
  Id_skrevet_paa)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Dok_Dew                               on Dok_Dew (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Nokkelord                             on Nokkelord (
  Id_bok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Create Index Dokument_Sammendrag                            on Sammendrag (
  Id_dok)
 Storage (INITIAL 4K NEXT 4K
     PCTINCREASE 1)
/

Rem ***  Domain Validations

Rem ***
Rem ***  Function procedures for the Data Model
Rem ***

Rem ***
Rem ***  Packages (with accessor functions) used for reselecting rows...
Rem ***

CREATE OR REPLACE PACKAGE Ans_Kor_Pack AS
  Id_dok Ans_Korp.Id_dok%TYPE;
  Korp_id_ansvarlig Ans_Korp.Korp_id_ansvarlig%TYPE;
  Koden_for Ans_Korp.Koden_for%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Ans_Per_Pack AS
  Id_bok Ans_Pers.Id_bok%TYPE;
  P_Id_er Ans_Pers.P_Id_er%TYPE;
  Koden_for Ans_Pers.Koden_for%TYPE;
  mutating BOOLEAN;
```

```
END;
/

CREATE OR REPLACE PACKAGE Bokspra_Pack AS
  Spraak_brukt_i Bokspraak.Spraak_brukt_i%TYPE;
  Id_skrevet_paa Bokspraak.Id_skrevet_paa%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Dok_Dew_Pack AS
  Id_dok Dok_Dew.Id_dok%TYPE;
  Deweykode_dew_kode Dok_Dew.Deweykode_dew_kode%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Dokumen_Pack AS
  Id_for Dokument.Id_for%TYPE;
  ISSN_nr_issn Dokument.ISSN_nr_issn%TYPE;
  Aarstall_vert Dokument.Aarstall_vert%TYPE;
  Tid_reg_dato Dokument.Tid_reg_dato%TYPE;
  Info_tittel Dokument.Info_tittel%TYPE;
  Navnet_tittel Dokument.Navnet_tittel%TYPE;
  Lengde_for Dokument.Lengde_for%TYPE;
  Rel_nr_nr Dokument.Rel_nr_nr%TYPE;
  Tittelen_paa Dokument.Tittelen_paa%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Korpora_Pack AS
  Korp_id_for Korporasjon.Korp_id_for%TYPE;
  Navnet_korp_navn Korporasjon.Navnet_korp_navn%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Nokkelo_Pack AS
  Id_bok Nokkelord.Id_bok%TYPE;
  Ordet_ord Nokkelord.Ordet_ord%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Person_Pack AS
  P_Id_for Person.P_Id_for%TYPE;
  Navnet_paa Person.Navnet_paa%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PACKAGE Sammend_Pack AS
  Id_dok Sammendrag.Id_dok%TYPE;
  Teksten_sammendrag Sammendrag.Teksten_sammendrag%TYPE;
  mutating BOOLEAN;
END;
/

CREATE OR REPLACE PROCEDURE Init_Packs IS
BEGIN
```

```
  Ans_Kor_Pack.mutating := FALSE;
  Ans_Per_Pack.mutating := FALSE;
  Bokspra_Pack.mutating := FALSE;
  Dok_Dew_Pack.mutating := FALSE;
  Dokumen_Pack.mutating := FALSE;
  Korpora_Pack.mutating := FALSE;
  Nokkelo_Pack.mutating := FALSE;
  Person_Pack.mutating := FALSE;
  Sammend_Pack.mutating := FALSE;
END;
/

@gruppering_userprocs.sql

Rem ***
Rem ***  Table triggers for the Data Model
Rem ***

Rem ***  Delete triggers

Rem ***  Insert/update triggers


Rem ***  End Data Model
```